

**Luiz Alfredo Soares Garcindo**

**UMA ABORDAGEM SOBRE O USO DA HIPERMÍDIA ADAPTATIVA  
EM AMBIENTES VIRTUAIS DE APRENDIZAGEM**

Tese apresentada ao Programa de Pós-Graduação em  
Engenharia de Produção da Universidade Federal de Santa  
Catarina como requisito parcial para a obtenção do título de  
Doutor em Engenharia de Produção.

**Orientador: João Bosco da Mota Alves, Dr.**

Florianópolis  
2002

**Luiz Alfredo Soares Garcindo**

**UMA ABORDAGEM SOBRE O USO DA HIPERMÍDIA ADAPTATIVA  
EM AMBIENTES VIRTUAIS DE APRENDIZAGEM**

**Esta Tese foi julgada adequada para a obtenção do título de Doutor em Engenharia  
de Produção no Programa de Pós-Graduação em Engenharia da Produção da  
Universidade Federal de Santa Catarina.**

**Florianópolis, 06 de março de 2002**

**Professor Ricardo de Miranda Barcia, PH. D.**

**Coordenador do Curso**

**Banca Examinadora:**

  
\_\_\_\_\_  
**Prof. João Bosco da Mota Alves, Dr. Eng.**

**Orientador**

  
\_\_\_\_\_  
**Luiz Carlos Duclos, PH.D.**

**Examinador Externo**

  
\_\_\_\_\_  
**Renato Antônio Rabuske, Dr.**

  
\_\_\_\_\_  
**Vladilen dos Santos Vilar, Dr.**

**Examinador Externo**

  
\_\_\_\_\_  
**Luiz Fernando Jacinto Maia, Dr.**

**Moderador**

**Para Góia e nossos filhos Tiago, Lucas, Marcos e Luiza.**

## **Agradecimentos**

Aos Professores João Bosco da Mota Alves e Luiz Fernando J. Maia, pela amizade, incentivo e apoio em todas as horas.

Ao Professor Paulo Renato Bastos Pinto, amigo incansável, incentivador e debatedor que deu muitas contribuições no assunto da Tese.

Aos Professores da Banca, Renato A. Rabuske, Vladilen dos Santos Villar e Luiz Carlos Duclos, pelas observações críticas que muito contribuíram com o trabalho.

Aos acadêmicos Cristiano Ferreira, Fernando José Cardoso de Sá, Marcos B. Garcindo e Felipe Ribeiro Cardoso (in memoriam) pela grande contribuição no desenvolvimento de software.

Aos meus filhos Marcos, Lucas e Tiago pelo incentivo e ajuda na organização do material.

À Universidade do Sul de Santa Catarina – Unisul, pela oportunidade oferecida.

A todos aqueles de alguma forma colaboraram com o desenvolvimento deste trabalho.

**A Deus, pela luz.**

## **Resumo**

Sistemas de Hipermídia Adaptativa (SHA) constituem uma área de pesquisa recente com grande potencial de aplicação no desenvolvimento de ambientes de aprendizagem na Web.

A idéia desses sistemas é de personalizar ou adaptar um Sistema de Hipermídia (SH) para usuários. Assim cada usuário terá uma determinada visão do conteúdo e da estrutura de navegação, ou seja, o sistema tenta antecipar as necessidades e desejos dos usuários a partir de modelos que representam o seu perfil, nível de conhecimento e preferências.

Para alcançar esses objetivos um SHA torna-se uma peça de software complexa que requer a busca de soluções tecnológicas que conduzam a maior flexibilidade e proximidade com os objetivos dos usuários, no caso professores e alunos.

No presente trabalho faz-se uma abordagem sobre o uso de sociedade de agentes e de autômatos finitos com saída em Sistemas de Hipermídia Adaptativa. É proposta uma arquitetura geral para intensificar aspectos pedagógicos desejáveis no ambiente, obter facilidades na utilização e reutilização do material didático hipermídia, alcançar maior flexibilidade no controle dos elementos que compõem o sistema e ainda facilitar a modelagem dos elementos componentes do esquema de adaptação.

O mecanismo de adaptação previsto funciona integrado ao modelo do usuário e é definido por regras de produção, o que permite flexibilidade na definição das condições de adaptabilidade. Trabalhos futuros são propostos na área de educação.

**Palavras chave:** hipermídia, hipermídia adaptativa, autômatos com saída, ambientes de aprendizagem, sociedade de agentes.

## **Abstract**

Adaptive Hypermedia Systems constitutes a recent research area that shows great application capacity for the development of Web Learning Environment. The system's idea is to personalize or to adapt a Hypermedia System for users. Thus each user will have one determined vision of the content and the navigation structure, or either, the system tries to anticipate the user's necessities and desires from models that represent its profile, level of knowledge and preferences. To reach these objectives an Adaptive Hypermedia System becomes a complex part of software that requires the technological brainstorming that leads biggest flexibility and proximity with the user's objectives, professors and students in this case. The approach in this current work makes use of an agent's society and finite automatons with output, in Adaptive Hypermedia Systems. A general architecture is proposal to intensify desirable pedagogical aspects in the environment, to get easiness in the use and reuse of the hypermedia didactic material, to reach greater flexibility in the elements control that compose the system and still facilitate the component elements modeling. The foreseen mechanism of adaptation functions integrated to the model of the user and is defined by production rules, which allows flexibility in the definition of the adaptability conditions. Future works are proposed in the education area.

**Words key:** hypermedia, adaptive hypermedia, automatons with output, environments of learning, society of agents.

## Sumário

Resumo .....	5
Abstract .....	6
Lista de figuras .....	10
Lista de tabelas .....	12
Lista de abreviaturas .....	13
 1 INTRODUÇÃO.....	14
1.1 Considerações iniciais .....	14
1.2. Objetivos .....	16
1.3. Justificativa .....	17
1.4 Delimitações do trabalho .....	17
1.5 Estrutura do Trabalho .....	17
2 HIPERTEXTO E HIPERMÍDIA .....	19
2.1 Uma visão histórica de hipertexto e hipermídia .....	19
2.2 Conceituação básica de Hipertexto e Hipermídia .....	20
2.3 Hipermídia e educação .....	24
2.4 Conclusão do capítulo .....	26
3 SISTEMAS DE HIPERMÍDIA ADAPTATIVA .....	28
3.1 Introdução .....	28
3.2 Elementos de sistemas de hipermídia adaptativa .....	30
3.2.1 O modelo do domínio .....	30
3.2.2 O modelo do usuário .....	33
3.2.3 Modelo de adaptação .....	35
3.3 Espaço de adaptação .....	35
3.4. Métodos da apresentação adaptativa .....	36
3.5 Técnicas da apresentação adaptativa .....	38
3.6 Métodos da navegação adaptativa .....	40
3.7 Técnicas da navegação adaptativa .....	42
3.8 Exemplos de sistemas de hipermídia adaptativa aplicados na educação.....	45
3.8.1 Elm-Art .....	45
3.8.2 Interbook .....	47
3.8.3 Tangow .....	47
3.8.4 PT .....	48
3.8.5 AHA .....	48
3.8.6 Olac e Póla .....	49
3.8.7 EPI-UMOD .....	49
3.8.8 KBS-Hiperbook .....	50
3.9 Conclusão do capítulo .....	51
4 UTILIZANDO CONCEITOS DE LINGUAGENS FORMAIS EM SHA .....	53
4.1 Introdução .....	53
4.2 Autômatos e hipertextos como autômatos – visão teórica .....	54
4.3 Conclusão do capítulo .....	58
5 AGENTES .....	60
5.1 Introdução .....	60
5.2. A inteligência artificial distribuída (IAD) .....	61
5.3 Definição de agentes .....	62

5.4 Propriedades dos agentes .....	64
5.5. Arquiteturas de agentes .....	66
5.6. A Sociedade de agentes .....	67
5.6.1. Arquitetura quadro negro (blackboard) .....	67
5.6.2. Arquitetura de troca de mensagens .....	68
5.6.3. Arquitetura federativa .....	68
5.7 Agentes e o processo pedagógico .....	68
5.8. Comunicação entre agentes .....	69
5.9 Conclusão do capítulo .....	71
6 INTELIGÊNCIA ARTIFICIAL APLICADA NA EDUCAÇÃO (IAED) .....	72
6.1 Introdução .....	72
6.2. Histórico .....	73
6.3 Questões relevantes na área de IAED .....	74
6.3.1 Natureza do conhecimento .....	74
6.3.2 Natureza do aprendizado .....	75
6.3.3 Estilos de ensino .....	76
6.3.4 Novas tecnologias na educação .....	77
6.3.5 Medidas de eficácia de um sistema .....	77
6.4 Sistemas Tutores Inteligentes .....	77
6.4.1 Introdução .....	77
6.4.2. Conceituação .....	78
6.4.3 Componentes .....	78
6.4.4 Módulo modelo do aluno .....	79
6.4.5 Módulo pedagógico .....	79
6.4.6 Módulo domínio .....	79
6.4.7. Módulo de comunicação .....	80
6.4.8 Módulo especialista .....	80
6.5 Conclusão do capítulo .....	80
7. A ARQUITETURA PROPOSTA .....	82
7.1. Introdução .....	82
7.2 Autômatos e a representação do modelo do domínio .....	82
7.2.1 Conceitos e relacionamentos entre conceitos .....	83
7.2.2 Utilização de autômatos finitos com saída na representação do modelo do domínio .....	85
7.2.3 Interconexão entre o modelo do domínio e páginas hipermídia .....	88
7.3 Sociedade de agentes .....	90
7.4 Descrição geral de uma arquitetura unificada para a abordagem .....	92
7.5 O agente de interface .....	95
7.6 O agente de modelagem do usuário .....	95
7.6.1 O Modelo do Usuário – estrutura básica .....	95
7.6.2 Avaliação do conhecimento .....	97
7.6.3 Objetivos do usuário, história e preferências .....	100
7.7 O agente de adaptação .....	101
7.8 Arquitetura Computacional para o SHA .....	106
7.8.1 Arquitetura Cliente Servidor em três camadas .....	106
7.8.2 Arquitetura Computacional Proposta para o SHA .....	109
7.9 Visão geral sobre o sistema de hipermídia adaptativa protótipo .....	111
7.9.1 Introdução .....	111
7.9.2 Objetivos .....	112
7.9.3 Características gerais do sistema Uniclick 2000 .....	112



7.9.4 Considerações gerais sobre o sistema protótipo .....	114
8. CONCLUSÕES FINAIS .....	116
8.1 Contribuições do trabalho .....	118
8.2 Trabalhos futuros .....	119
REFERÊNCIAS .....	120
APÊNDICE A - AMBIENTE DE APRENDIZAGEM BASEADO EM HIPERMÍDIA ADAPTATIVA – PROTÓTIPO .....	129

## Lista de figuras

Figura 1: Hipertexto – exemplo de documento hipertexto .....	21
Figura 2: Hierarquia de conceitos (DE BRA; HOUBEN; WU, 1999) .....	31
Figura 3: Visão geral de um SHA .....	33
Figura 4: tecnologias de adaptação em hipermídia adaptativa clássica (Brusilovsky, 1996) .....	36
Figura 5: Modelo de segmento de um curso – máquina de Moore ( MACHADO et al., 1999).....	55
Figura 6: Modelo de Agentes (CAGLAYAN; HARRISON, 1997) .....	63
Figura 7: Componentes Básicos de um Sistema Tutor Inteligente (BECK et al., 1998) .....	79
Figura 8: hierarquia de conceitos – modelo de referência AHAM (De BRA; HOUBEN; WU, 1999) .....	83
Figura 9: Autômato mapeado (máquina de Moore) .....	88
Figura10: Indexação por página (BRUSILOVSKY; PESIN, 1994) .....	89
Figura 11: Indexação fragmentada (BRUSILOVSKY; PESIN, 1994) .....	89
Figura 12: Relacionamento direto (BRUSILOVSKY; PESIN 1994) .....	90
Figura 13: Arquitetura geral proposta na abordagem (rede de estações/canais) .....	93
Figura 14: Avaliação do conhecimento através de coeficientes de confiança .....	99
Figura 15: Propagação do conhecimento em estrutura de conceitos .....	100
Figura 16: Arquitetura cliente servidor com 2 camadas .....	107
Figura 17: Arquitetura cliente servidor com 3 camadas (PEREIRA, 1999) .....	109
Figura 18: Arquitetura computacional para SHA, em 3 camadas utilizando Ambiente JAVA .....	110
Figura 19: Interface inicial do sistema Uniclick2000 .....	134
Figura 20: Interface para visualizar matrículas em cursos .....	135
Figura 21: Interface mostrando sumário do material instrucional para o aluno. ....	136
Figura 22: Interface para o professor autor na criação e manutenção de páginas, autômatos e estrutura do curso .....	138
Figura 23: Diagrama ER Máquina de Moore .....	143

.Figura 24: Diagrama ER – Máquina de Mealy .....	144
Figura 25: Arquitetura computacional utilizada no protótipo .....	145

## **Lista de tabelas**

Tabela 1 - Métodos e Técnicas de Apresentação Adaptativa em SAH (BRUSILOVSKY, 1996) .....	40
Tabela 2 - Métodos e Técnicas de Navegação Adaptativa em Sistemas de Hipermídia Adaptativa (BRUSILOVSKY, 1996) .....	45
Tabela 3 - Tabela do modelo do usuário (De BRA; HOUBEN; WU, 1999).....	96
Tabela 4 – Módulos do sistema Uniclick2000 .....	132
Tabela 5 – Níveis de autorização do sistema Uniclick2000 .....	134

### Lista de abreviaturas

SHA	Sistema de hipermídia adaptativa
SH	Sistema de hipermídia
IA	Inteligência artificial
IAD	Inteligência artificial distribuída
IAED	Inteligência artificial na educação
HA	Hipermídia adaptativa
AHAM	Adaptive hypermedia application model
HDM	Hypermedia design model
WWW	World Wide Web
EA	Explicação adicional
ER	Explicação requerida
EC	Explicação condicional
EV	Explicação variante
TC	Texto condicional
FV	Fragmento variante
PV	Página variante
CG	Condução global
CL	Condução local
OL	Orientação local
OG	Orientação global
OD	Orientação direta
MA	Mapas adaptativos
STI	Sistema tutor inteligente
MAS	Sistemas multiagentes
DSP	Resolução distribuída de problemas
KQML	Knowledge query and manipulation language
FAC	Frequently asked questions

# **1 INTRODUÇÃO**

## **1.1 Considerações Iniciais**

O uso de recursos da Internet na educação e treinamento tem atraído a atenção de pesquisadores e educadores, de um modo geral, em todo o mundo.

Fatores diversos, de um modo especial as facilidades de recuperação da informação e as amplas possibilidades de interatividade, tornam visível a capacidade do uso dos recursos da grande rede na área educacional. Os benefícios são claros, pois é possível atender às necessidades de milhares de estudantes, independente de tempo e lugar, a um custo mais baixo.

Neste contexto são viabilizadas novas formas de aprendizagem através de novos recursos didático-pedagógicos e de novas tecnologias. Isso propicia a criação de ambientes de aprendizagem na Internet, voltados para o ensino à distância ou para o apoio ao ensino presencial.

A Hipermissão Adaptativa é uma área de pesquisa recente com ótimas perspectivas para a criação de ambientes virtuais de aprendizagem. Brusilovsky (1996, p. 2) define um Sistema de Hipermissão Adaptativa (SHA) como: “ Denomina-se Sistema de Hipermissão Adaptativa todo o sistema baseado em hipertexto e hipermissão que reflita algumas características do usuário em modelos e aplique tais modelos na adaptação de diversos aspectos visíveis do sistema ao usuário”.

Nesse enfoque cada usuário terá uma determinada visão do conteúdo e da estrutura de navegação, ou seja, o sistema tenta antecipar as necessidades e os desejos de seus usuários, a partir de modelos que representam o seu perfil, o seu nível de conhecimento e suas preferências.

Na sua concepção, os SHA aproveitam conceitos oriundos de Sistemas de Hipertexto, Sistemas de Hipermídia e Sistemas Tutores Inteligentes, que foram desenvolvidos nas décadas de 80 e 90. Os primeiros SHA para uso na Web e aplicados na educação, foram relatados a partir de 1996 (BRUSILOVSKY et al., 1996).

A arquitetura desses sistemas envolve basicamente três partes: o modelo do domínio, o modelo do aluno e o esquema de adaptação ou módulo de adaptação.

**O modelo do domínio** é a base de todo o sistema, pois descreve como a informação de conteúdo é estruturada. No ensino a distância pela Web o modelo do domínio é freqüentemente baseado no currículo seqüencial, no qual os conceitos, exemplos, testes, tarefas, etc., são organizados em tópicos e subtópicos da maneira mais apropriada para o usuário.

**O modelo do usuário** tem como base o seu perfil e as suas ações. Para tanto o sistema coleta, inicialmente, informações como preferências, experiência anterior, metas e outros aspectos de interesse. Por outro lado, monitorando o comportamento do usuário na navegação, o sistema mantém informações sobre o seu nível de conhecimento em relação aos diversos conceitos abordados na aplicação.

**O esquema de adaptação** é baseado num conjunto de regras de adaptação (regras pedagógicas) e de mecanismos que realizam a adaptação em função dessas regras. As regras definem como a adaptação deve ocorrer.

**O problema** que se apresenta é que para alcançar seus objetivos, um SHA torna-se uma peça de software complexa, que requer a busca de soluções tecnológicas que conduzam a maior flexibilidade, modularidade e proximidade com os objetivos dos usuários. No caso de aplicações de SHA na área educacional, em ambientes de aprendizagem, é desejável o uso de recursos da engenharia de *software* que permitam:

- aos usuários alunos uma navegação assistida mais efetiva, buscando intensificar aspectos pedagógicos desejáveis no ambiente;
- aos usuários autores, responsáveis pela preparação dos cursos no ambiente, facilidades relacionadas à construção do modelo do domínio no sistema, principalmente nos seguintes aspectos:
  - a criação do material instrucional hipermídia (páginas) da forma mais independente possível da estrutura de controle de navegação do sistema como um todo (MAURER, 1997);

- a reutilização das páginas em diversos cursos com a eliminação de redundâncias;
- a elaboração de seqüências instrucionais com enfoques diferenciados dentro de um mesmo curso;
- maior flexibilidade no controle dos elementos que compõem o sistema, modularidade, expansibilidade e melhor adaptabilidade em estruturas computacionais distribuídas;
- facilidades para definir as condições de adaptação a partir de requisitos pedagógicos.

Em geral, as soluções conceituais e tecnológicas para os SHA são apresentadas de forma não unificada. Ora prioriza-se os sistemas de navegação, ora aspectos conceituais de modelagem de cursos, alunos ou formas diferentes de implementações na Web.

Neste trabalho busca-se uma solução unificada para uma arquitetura de SHA. A solução envolve o uso de sociedade de agentes no esquema de adaptação como um todo e autômatos finitos com saída, na estruturação e controle da base de hiperdocumentos (modelo do domínio).

## 1.2. Objetivos

Este trabalho tem como objetivo geral fazer uma abordagem sobre a unificação dos conceitos de sociedade de agentes e autômatos finitos com saída em SHA, definindo uma arquitetura geral para o processo de adaptação.

Os objetivos específicos são:

1. A revisão da literatura sobre hipermídia, hipertexto, sistemas de hipermídia, sistemas de hipermídia adaptativa, agentes, sistemas tutores inteligentes, e inteligência artificial aplicada na educação.
2. Apresentação de uma arquitetura geral para o esquema de adaptação, englobando a unificação da tecnologia de agentes e conceitos de autômatos finitos com saída.
3. O projeto e desenvolvimento de um sistema de hipermídia protótipo, para ambiente Web, aplicado à educação, utilizando autômatos finitos com saída para a navegação e estruturação do modelo do domínio.



4. O estudo e definição de uma arquitetura computacional distribuída para suportar SHA.

### **1.3. Justificativa**

A tendência de utilização de SHA é muito forte e pode ser constatada em publicações recentes. A idéia principal destes sistemas é resolver problemas de navegação e a apresentação da informação mais relevante para o usuário, utilizando técnicas da inteligência artificial. A hipermídia está presente na Internet e em aplicações voltadas para a recuperação de informações de um modo geral. No caso da área de educação, ambientes de aprendizagem inteligentes, baseados em hipermídia, constitui-se em ferramentas importantes, principalmente considerando a expansão do ensino a distância via Internet, que está ocorrendo no mundo inteiro. Constata-se também que no Brasil esta área desperta a atenção de educadores e pesquisadores, porém é carente em publicações e projetos.

Este trabalho pretende estudar e apresentar soluções para SHA aplicados à educação e dessa forma contribuir para o desenvolvimento da educação no País.

### **1.4 Delimitações do trabalho**

Não é intenção do trabalho abordar todos os aspectos de um ambiente de aprendizagem para utilização na WEB, mas sim, realizar um estudo específico sobre o esquema de adaptação em SHA, através da unificação da tecnologia de agentes e conceitos de autômatos finitos com saída. A abordagem utiliza como referência diversas pesquisas e modelos propostos na área de hipermídia adaptativa, bem como trabalhos sobre o uso de autômatos em sistemas de hipermídia aplicados na educação.

### **1.5 Estrutura do Trabalho**

O presente trabalho está estruturado em oito capítulos e um apêndice.

Os capítulos dois e três são revisões dos assuntos hipertexto, hipermídia e hipermídia adaptativa, incluindo técnicas de navegação e apresentação adaptativas.

O capítulo quatro trata dos conceitos sobre autômatos, hipertexto como autômato e aplicações de autômatos em sistemas de hipermídia.

O capítulo cinco trata do conceito de agentes e de arquiteturas baseadas em agentes.

O capítulo seis aborda a inteligência artificial aplicada na educação

O capítulo sete apresenta a arquitetura proposta e informações gerais sobre um sistema protótipo construído.

O capítulo oito apresenta a conclusão da pesquisa realizada, principais contribuições e recomendações para futuros trabalhos.

No Apêndice A são apresentadas informações técnicas sobre o sistema hipermídia implementado como protótipo.

## **2 HIPERTEXTO E HIPERMÍDIA**

### **2.1 Uma visão histórica de hipertexto e hipermídia**

As primeiras idéias sobre mecanismos para armazenamento e manipulação de informações de forma não linear, foram creditadas para Vannevar Bush quando escreveu o artigo “*As We May Think*” (BUSH, 1945), no qual propôs uma máquina imaginária denominada Memex para armazenamento e processamento de informações. Bush era formado em Engenharia Elétrica e na época gerenciava um Instituto de Pesquisa e Desenvolvimento do Governo Norte Americano. Uma de suas preocupações era de armazenar de forma adequada todo o conhecimento científico produzido, para possibilitar a fácil consulta pela comunidade por ele gerenciada, obtendo dessa forma ganhos em produtividade.

O sistema Memex, foi idealizado originalmente como uma ferramenta de acesso a informações armazenadas em microfilmes e fotocélulas, na qual os usuários poderiam criar trilhas de informações através de ligações entre textos e ilustrações correlacionadas. A máquina facilitaria a vida das pessoas na convivência com uma crescente demanda de informações. Bush visualizou o conceito mas jamais concretizou a criação da máquina imaginada (CHAIBEN, 2001).

Nos anos 60, Douglas Engelbart (ENGELBART, 1963), pesquisou os conceitos envolvidos nas idéias de Bush, com o objetivo de utilizar computadores para aumentar a capacidade e produtividade humanas.

Através do projeto *Augment*, vinculado ao Instituto de Pesquisa de Stanford, Engelbart, seguindo idéias próprias, desenvolveu diversas técnicas e instrumentos que hoje são comuns, tais como: o mouse, janelas múltiplas numa tela, sistema de ajuda integrados, correio eletrônico.

No contexto do Projeto *Augment*, Engelbart desenvolveu o sistema NLS (*On-Line System*), voltado para o armazenamento de documentos interrelacionados. Esse sistema pode ser considerado o primeiro hipertexto construído. Mais tarde, com posteriores desenvolvimentos o NLS se tornou o Sistema de Hipertexto do Projeto *Augment*, tendo sido comercializado pela Empresa McDonnell-Douglas.

Em 1965, Ted Nelson baseado nos trabalhos já realizados, criou o termo “hipertexto”. Sua visão envolvia um grande repositório de informações literárias não deletáveis, com acesso aberto para todos os escritores do mundo, que poderia ser navegado de forma não linear, ficando a critério do navegador a escolha dos links. Isso era mais que um texto, era o hipertexto. Em 1967, Nelson colocou suas idéias no projeto Xanadu, um sistema de hipertexto de alto desempenho, capaz de armazenar na forma eletrônica documentos literários de todo o mundo. Neste projeto Nelson antecipou a visão sobre o impacto que a tecnologia do hipertexto produziria na sociedade futura. Atualmente a Web realiza parte dessa visão, exceto que as informações podem estar em mais de um lugar e podem ser excluídas. O projeto de um servidor de rede universal para o Xanadu foi descrito em várias edições de seu livro *Literary Machines* (NELSON, 1993).

As previsões de Nelson tornam-se realidade com o advento da Internet. O conceito de hipertexto desempenha para a *World Wide Web* (WWW ou simplesmente Web) um papel de fundamental importância, pois permite estabelecer, para a grande rede, uma nova dimensão em termos de organização, apresentação e acesso à informação.

A pesquisa em hipertexto foi definida como um campo de investigação a partir do desenvolvimento dos sistemas *Guide* em 1986 e *Hypercard* em 1987, e a realização da primeira conferência sobre o assunto organizada pela ACM neste último ano.

## 2.2 Conceituação básica de Hipertexto e Hipermissão

Originalmente hipertexto é um conjunto de nós de textos conectados por links. Cada nó contém alguma quantidade de informação (texto) e um número de links para outros nós. A figura a seguir mostra um exemplo de um documento hipertexto.

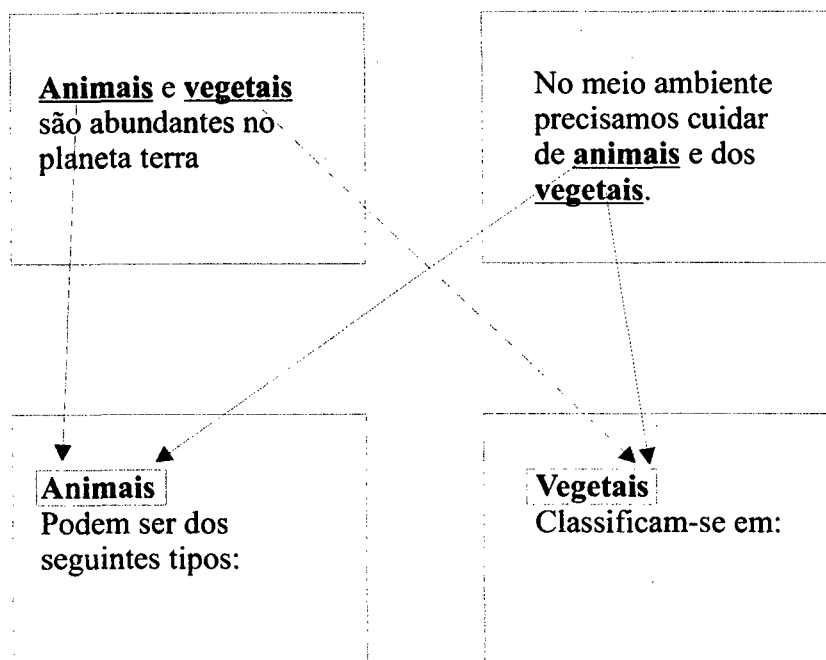


Figura 1 – Hipertexto – exemplo de documento hipertexto

Essa estrutura de nós e links na base de conhecimento hipertexto, possibilita ao usuário percorrer um espaço de informação utilizando as chamadas ferramentas de navegação. A estrutura pode ser vista como um grafo e pode assumir uma complexidade arbitrária.

Na figura 1 apresenta-se um pequeno hiperdocumento com quatro nós e quatro links. Percebe-se que os links apontam para pontos específicos, palavras ou regiões no interior de um nó que denominam-se âncoras. Para melhor entendimento, são apresentados a seguir alguns conceitos (SANTOS, 1996):

- **hiperdocumento:** pode ser visto como um banco de dados, organizado como uma estrutura de rede, onde os nós, unidos por links, contém trechos de informação;
- **nó:** é a menor unidade de informação e contém um trecho de informações definidas pelo autor. Um nó é normalmente associado a uma janela exibida na tela, e sua manipulação tenta imitar o manuseio de livros. Um nó normalmente representa um conceito e pode conter texto, gráficos, animação, áudio, vídeo, imagens ou programas;

- **link:** representa o relacionamento definido pelo autor entre dois trechos de informação. Um nó é mostrado na tela com uma ou mais palavras marcadas (âncoras), através das quais o usuário ativa um link, abrindo uma janela de novas informações ou informações de origem.

Os links, tem como funções :

- associar idéias diferentes em diferentes nós;
  - conectar nós a outros nós;
  - conectar anotações e comentários aos textos a que pertencem;
  - oferecer informação organizacional, como por exemplo, estabelecer relações entre duas partes de textos ou entre o índice de conteúdo do texto e suas seções;
  - conectar informações explicativas e documentos a um mapa, figura ou gráfico, auxiliando seu entendimento.
- **caminhos:** é uma seqüência de links definida pelo autor do hiperdocumento ou pessoa autorizada, para ser seguida por um usuário em uma leitura particular. Podem ser estabelecidos vários caminhos para um mesmo hiperdocumento, para que este possa ser apresentado de diversas formas;
  - **mapa global:** é a apresentação de nós em forma de um grafo, onde são indicados os links de todos os nós entre si. É possível em alguns hipertextos se fazer uma navegação por estes nós de forma gráfica. O problema encontrado em tal representação, reside no fato de que em grandes hiperdocumentos o mapa global, provavelmente, se tornará ilegível;
  - **mapa regional:** apresenta o grafo dos nós de forma regional, focalizando determinado número de informações;
  - **autoria:** é a propriedade que permite a criação do hipertexto que reflita a realidade do assunto em questão e satisfaça os requisitos da aplicação, podendo destacar aqueles elementos de informação mais relevantes e relacioná-los a outros no contexto do assunto que se desenvolve.

Sistemas de hipertexto são ferramentas de software complexas compostas de diversos componentes. Nos últimos anos foram propostos os seguintes modelos de referência para sistemas de hipertextos:

- modelo Trellis (STOTTS; FURUTA, 1989);
- modelo Dexter (HALASZ; SCHWARTZ, 1990);
- modelo formal de hipertexto( LANGE, 1990);
- modelo *Tower*, orientado a objetos, proposto por De Bra, Houben e Kornatzi (De BRA et al. 1992);
- *Ham ou Hypertext Abstract Machine*, (CAMPBELL, 1988).

Hipermídia é uma extensão do hipertexto. Para muitos, a palavra originou-se da junção de hipertexto com multimídia. Os termos Hipertexto e hipermídia são considerados atualmente como sinônimos (LOWE; HALL, 1999), pois a maioria dos sistemas de hipertexto atuais permite o uso de outras formas de representação, incluindo sons, gráficos, animação, vídeo, etc. Neste trabalho será utilizado preferencialmente o termo hipermídia.

Sistemas Hipermídia (SH), são sistemas que possibilitam o projeto e o acesso a documentos hipermídia, ou seja, fornecem métodos de acesso não sequencial a informações através da apresentação de nós e links.

Em Maidantchik e Xexéo (1996, p. 4), tem-se:

[...] essencialmente um sistema hipermídia é uma coleção de objetos heterogêneos – fragmentos cognitivos como textos, imagens, sons conectados entre si. Um sistema de hipertexto é feito de nós (conceitos) e links (relacionamentos). Um nó representa um simples conceito ou idéia. Links conectam conceitos relacionados ou nós. Pode-se navegar o grafo de conceitos posicionando no browser itens de interesse. A principal característica do hipertexto é a noção de links internos ou entre documentos suportados pela máquina.

Segundo Balasubramanian (1994), as características básicas dos sistemas hipermídia são: uma interface normalmente gráfica, com apoio de folheadores e uma visão geral de diagrama; um sistema de autoria com ferramentas para a criação e administração de nós e links; mecanismos tradicionais de recuperação de informação e uma máquina de hipermídia para administrar a informação contida nos nós e links. Nesses sistemas o armazenamento pode ser feito através de um gerenciador de arquivos, uma base de conhecimento, um sistema de gerenciamento de banco de dados relacional ou um sistema de gerenciamento de banco de dados orientado a objetos.

Os SH tornaram-se muito populares como ferramentas para guiar usuários na procura de informação. Oferecem muita liberdade na navegação no hiperespaço o que muitas vezes torna-se um problema. Por outro lado, esses sistemas não consideram normalmente as características do usuário, suas preferências e seus objetivos na navegação. O uso de hipermídia tornou-se popular na Internet através da Web (*World Wide Web*) concebida por Tim Bernes Lee em 1989. A Web é um grande sistema de hipermídia com suporte na Internet, que conta atualmente com milhões de usuários.

As principais tecnologias que dão suporte à Web como um sistema de hipermídia, podem ser divididas em:

- uma arquitetura de processamento distribuído do tipo cliente-servidor;
- um sistema de endereçamento denominado URI – *Uniform Resource Identifier*, que permite referências padronizadas a cada item da Web;
- uma Linguagem para representação de hiperdocumentos denominada Html - *Hypertext Mark-up Language* ;
- um protocolo de rede denominado *Http – Hypertext Transfer Protocol* , para a recuperação de informações na Web.

## 2.3 Hipermídia e educação

A tecnologia hipermídia é aconselhada para uso em ambientes educacionais computadorizados porque permite ao estudante a navegação livre sobre o material instrucional, representado em diversas mídias e estruturado em forma de nós e links. O controle de navegação fica com o estudante permitindo que ele evolua conforme seus interesses e objetivos. A característica pedagógica mais importante portanto é a flexibilidade de exploração do material didático (WHALLEY, 1993). Entretanto, a tecnologia hipermídia não foi projetada especialmente para a educação

O principal foco dessa tecnologia tem sido a recuperação eficiente da informação e o entretenimento. Tem-se então dois tipos de aplicações hipermídia (THÜRING et al., 1995):

- aplicações destinadas à pesquisa e recuperação de informações que apresentam-se como base de dados que podem ser exploradas livremente por um leitor;



- aplicações destinadas a tarefas que envolvem compreensão e aprendizagem com o formato de documentos eletrônicos, guiando intencionalmente os leitores através de um espaço de informações sobre um domínio de conhecimento.

Segundo Nadeau (apud LAROUSSI, 2000), a estrutura tipo hipertexto apresenta as seguintes vantagens num ambiente de aprendizado:

- favorece o pensamento associativo;
- suscita a iniciativa do aprendiz em interagir com o sistema;
- pode ser usado como suporte para a aprendizagem colaborativa, pois os estudantes podem utilizar um mesmo recurso e conjugar esforços para compreender ou resolver um problema dado;
- facilita a aprendizagem multidisciplinar, permitindo a construção de pontes entre diversos domínios do conhecimento.

Por outro lado há pelos menos dois aspectos críticos: navegação e autoria . Na navegação tem-se os seguintes aspectos críticos ( BALASUBRAMANIAN, 1994):

- desorientação, que é um efeito cognitivo produzido sobre o usuário que perde o link de seu plano de navegação com zonas de informação que está consultando. O problema pode surgir em função do usuário não lembrar que informações já recebeu ou então por não conseguir estabelecer a conexão entre informações obtidas e os objetivos perseguidos. É muito comum a frase decorrente “perdido no hiperespaço”;
- sobrecarga cognitiva, é o esforço e a concentração adicionais necessários para manter várias tarefas ou trilhas ao mesmo tempo. Como a capacidade humana de processamento de informação é limitada, cada esforço adicional para a leitura reduz os recursos mentais disponíveis para a compreensão. Em hiperdocumentos, esses esforços correspondem principalmente à orientação, navegação e adaptação à interface de usuário.
- sobrecarga de informação, resultado da diminuição da capacidade de atenção no assunto principal em função do grande número de informações recebidas. Ocorre a chamada “poluição de informações”.

Com relação à autoria, verifica-se que o processo de criação de cursos apresenta dificuldades na preparação do material para uso em ambientes hipermídia. Há a necessidade de soluções que facilitem a transformação do material instrucional em estrutura hipermídia. O professor autor sofre também com a sobrecarga cognitiva ao nomear nós e realizar links

semanticamente corretos entre esses nós. Para minimizar os efeitos adversos na autoria, diversos métodos de suporte tem sido propostos (SANTOS, 1996):

- HDM - *Hypermedia Design Model* (GARZOTTO et al., 1993);
- HDM2 - uma extensão de *Hypermedia Design Model* (GARZOTTO et al., 1995);
- OOHDM - *Objects Oriented Hypermedia Design Model*, uma extensão de *Hypermedia Design Model*, considerando os componentes de um hiperdocumento como objetos (SCHWABE; ROSSI, 1995);
- RMM - *Relational Management Model* (ISAKOWITZ et al., 1995);
- MacWeb, um modelo para especificação e projeto de hiperdocumentos a partir de nós tipados (NANARD E NANARD, 1995);
- Hiper-Autor (BREITMAN, 1993).

Há que se considerar também que a maioria dos documentos usados na área educacional, não deve ser lida numa ordem arbitrária porque o conhecimento é construído gradualmente e os primeiros capítulos contêm conceitos que são pré-requisitos de capítulos mais avançados. Dessa forma ocorre que muitas vezes um texto seqüencial transformado numa estrutura hierárquica com links, pode ocasionar a ilusão de que o leitor pode pular diretamente para qualquer capítulo ou seção que desejar. No entanto, relações de pré-requisitos entre seções ou capítulos, podem resultar no alcance de páginas que ele não pode entender ( De BRA, 1999).

## 2.4 Conclusão do capítulo.

Verificou-se nesse capítulo que a utilização do conceito de hipertexto/hipermídia tem evoluído muito rapidamente e traz uma série de vantagens nas diversas áreas de aplicações, com ênfase na área de educação. A popularização da Web nos últimos anos tem impulsionado as pesquisas na busca de soluções para os problemas comuns de usuários de sistemas de hipertexto/hipermídia, tais como, desorientação, sobrecarga cognitiva e sobrecarga de informações. Entre as soluções sobressai a navegação assistida, adaptada às necessidades do usuário, que envolve a integração de sistemas inteligentes ( sistemas baseados em conhecimento) com sistemas hipermídia, surgindo assim os denominados sistemas de hipermídia adaptativa (SHA).

Os SHA são softwares complexos porque envolvem diversas áreas do conhecimento. É preciso trabalhar com a modelagem do conhecimento, a modelagem do usuário e com técnicas de adaptação de interfaces. Além disso requerem normalmente arquiteturas sofisticadas para o seu funcionamento, baseadas em processamento distribuído. No próximo capítulo pretende-se fornecer uma visão mais ampla dos diversos componentes dos SHA.

### 3 SISTEMAS DE HIPERMÍDIA ADAPTATIVA

#### 3.1 Introdução

Há uma forte tendência pela utilização de sistemas baseados em hipermídia adaptativa na educação e em diversas áreas de negócios na Internet, como por exemplo, sistemas de informações pessoais, sistemas de ajuda *on-line*, sistemas de informações institucionais, entre outras. Esses sistemas são especialmente úteis em ambientes nos quais há uma diversidade de usuários com diferentes objetivos e níveis de conhecimentos. A proposta desses sistemas é de se adaptar às necessidades, objetivos e nível de conhecimento dos usuários. Pode-se dizer que a maioria dos sistemas de hipermídia adaptativa (SHA) existentes foi produzida em laboratórios de Universidades e é utilizada em pequena escala.

Na última década, diferentes tipos de sistemas de hipermídia e sistemas baseados na Web foram construídos de forma a introduzir alguma forma de personalização. Surgiram então os seguintes tipos de sistemas (De BRA, 1999):

- **sistemas de hipermídia adaptáveis** nos quais é obtido o perfil dos usuários através de questionários ou diálogos. Os sistemas oferecem uma versão da aplicação hipermídia correspondente ao perfil desses usuários. Dados do perfil podem incluir preferências de apresentação como cores, tipo de mídia, estilo da aprendizagem e história do usuário, ou seja, suas qualificações e seu conhecimento sobre determinados assuntos. Na Internet há diversos *sites* que solicitam o preenchimento de questionários com o objetivo de apresentar ao usuário informações relacionadas ao seu perfil;
- **sistemas de hipermídia adaptativa** nos quais o comportamento do usuário é monitorado e a apresentação hipermídia, previamente definida, é adaptada de forma

dinâmica de acordo com este comportamento. Esses sistemas podem também utilizar o recurso de questionários ou testes para aprimorar o conhecimento sobre o usuário;

- **sistemas de hipermídia dinâmicos**, nos quais o comportamento do usuário também é monitorado, com a diferença que ao invés de alterar uma apresentação previamente definida esses sistemas geram a apresentação a partir de itens de informação atômicos.

Os SHA constituem uma área de pesquisa que permitirá a consolidação da hipermídia. Esses sistemas constroem um modelo das metas, preferências e conhecimento de um determinado usuário e usam esse modelo para realizar a adaptação da hipermídia às suas necessidades.

Pode-se dizer que um SHA executa basicamente três funções (De BRA, BRUSILOVSKY, HOUBEN, 1999):

- enquanto o usuário está navegando todas as suas ações são registradas. A partir dessas ações o SHA mantém informações sobre o conhecimento do usuário em relação a cada conceito do domínio do conhecimento. Tipicamente no modelo do usuário são mantidos dinamicamente atributos que definem quanto o usuário sabe sobre determinado conceito (valor do conhecimento) e também o que foi lido;
- o modelo do usuário é usado para direcionar as ações de adaptabilidade, ou seja, as páginas (nós) são classificadas em grupos de acordo com o seu atual nível de conhecimento, suas metas e preferências. O SHA manipula então âncoras de links constantes dos nós (e links destino) para guiar o usuário na obtenção de informações relevantes. Dependendo da classe de um nó uma âncora de link pode ser anotada, desabilitada ou removida;
- o conteúdo de uma página de apresentação pode ser modificado para melhor atender as necessidades do usuário. Pequenas explicações podem ser adicionadas, comparações com assuntos descritos em outras páginas que o usuário viu anteriormente, detalhes adicionais para usuários avançados etc. Mudanças no estilo da apresentação, seleção de diferentes mídias (texto, imagem, som, vídeo), alteração do tamanho da apresentação, podem ocorrer considerando o perfil do usuário, realizando o que se chama de apresentação adaptativa.

As formas de adaptação serão discutidas mais adiante no contexto deste trabalho.

## 3.2 Elementos de sistemas de hipermídia adaptativa

Diversos modelos para SHA foram desenvolvidos. De Bra, Houben e Wu (1999), apresentam um modelo de referência para sistemas de hipermídia adaptativa (AHAM), como extensão do modelo de referência Dexter (HALASZ; SCHWARTZ, 1990) para sistemas de hipermídia. Neste modelo três elementos fortemente relacionados são identificados: o modelo do domínio, o modelo do usuário e o esquema de adaptação. A seguir descreve-se sucintamente estes elementos.

### 3.2.1 O modelo do domínio

O modelo do domínio indica como o conteúdo da informação (hiperdocumento) está estruturado. Há três níveis, conforme figura 2:

- **fragmentos** – é o nível mais baixo constituído de fragmentos de informação. São consideradas unidades atômicas. Pode ser um parágrafo de texto, uma imagem, um videoclipe etc. O SHA não explora o interior de um fragmento. Fragmentos podem ser unidades estáticas de texto ou podem ser gerados por softwares específicos;
- **nós, ou páginas** - são unidades de apresentação para o usuário. A página é construída à parte e incluirá fragmentos de acordo com o mecanismo de adaptação;
- **conceitos compostos** – o domínio da aplicação pode ser descrito em termos de conceitos de alto nível. A relação entre os conceitos pode existir para indicar caminhos de navegação desejáveis. Alguns conceitos fazem parte de um conceito maior numa concepção hierárquica. Pode haver uma relação do tipo um a um entre conceitos e nós (granulação fina), no entanto um conceito poderá envolver uma série de nós.

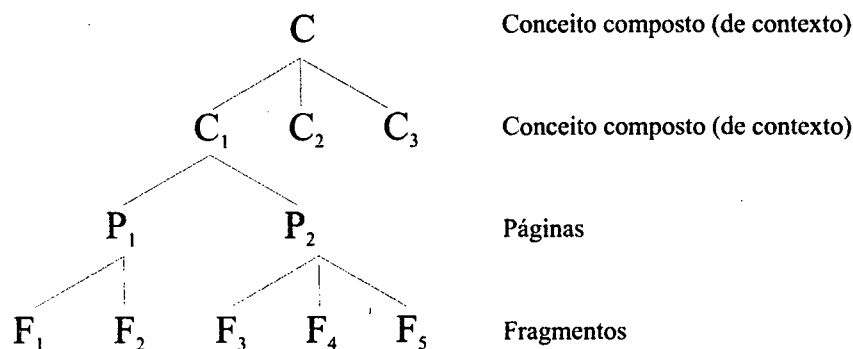


Figura 2 - Hierarquia de conceitos (De BRA, HOUBEN E WU, 1999).

Neste trabalho tem-se como base o paradigma do currículo seqüencial para representar o domínio do conhecimento, considerando as características esperadas para ambientes de aprendizagem que possam funcionar na Web em aplicações de ensino à distância.

No currículo seqüencial o conhecimento é organizado num conjunto de conceitos ou tópicos, tarefas como exemplos, exercícios e problemas, numa seqüência planejada para permitir o aprendizado do estudante da melhor maneira possível. Os tópicos representam pedaços elementares de conhecimento para um dado domínio e o tamanho do tópico depende do domínio.

Alguns sistemas baseiam-se em modelos de domínio mais ou menos complexos. Segundo Santos (1996), os modelos de domínio podem ser classificados em:

- modelos simples, chamados também de **modelos um**, não usam os nós tipados;
- modelos mais avançados diferenciam várias formas de tópicos e muitas formas de links no modelo de domínio, sendo este modelo constituído como uma rede semântica. Esses modelos são chamados de **modelos dois**;
- finalmente, nos **modelos três** são utilizados frames para representar as estruturas internas dos tópicos através de um conjunto de atributos.

A estrutura do domínio do conhecimento pode ser montada de diversas formas, como visto anteriormente. Segundo De Bra (1999), o modelo do domínio é definido em três níveis (conceito, página e fragmentos) e deve ser convertido para um nível, tendo em vista que na Web a unidade de apresentação é a página (página Web). Algumas formas de se fazer isso são sugeridas pelo autor conforme os casos a seguir:

**Caso 1** - em Silva et al.(1998), é descrito um SHA aplicado à educação no qual uma ferramenta gráfica é usada para descrever como diferentes páginas são associadas com um simples conceito do domínio. Cada página contribui com uma fração do conhecimento do conceito. Nesses casos podem ocorrer situações de difícil implementação, tais como:

- quando uma página é lida mais de uma vez, sua contribuição para o conceito não deve ser contada duas ou mais vezes;
- quando uma página é lida com pré-requisitos não satisfeitos, a contribuição para o conhecimento do conceito será provavelmente menor do que no caso de leitura quando a página está pronta para a leitura;
- duas páginas podem sobrepor informação e então ocorre sobreposição sobre o conhecimento do conceito. Não fica claro qual contribuição deve ser considerada;
- a soma das contribuições em páginas sobre um conceito pode exceder 100% mas o conhecimento não pode exceder 100%.

**Caso 2** - em sistemas como Interbook, o domínio da aplicação é descrito por um documento estruturado. Seções do documento são associadas com conceitos do domínio. Um procedimento traduz o documento em páginas html e associa conceitos às páginas. Cada página tem conceitos anteriores (pré-requisitos) e conceitos resultados. O conhecimento sobre conceitos resultados é gerado pela leitura da página.

**Caso 3** - o Sistema AHA associa zero ou mais conceitos ( tipicamente um) a uma página. Na versão corrente o conhecimento sobre um conceito é gerado somente se a página é lida quando os pré-requisitos estão satisfeitos. Ao invés de usar um conjunto de conceitos como conhecimento pré-requisito, cada página conta com requisito que é uma expressão *booleana* nos conceitos. Através do uso de *and* e *or* e arbitrários parênteses uma rica coleção de requisitos pode ser formulada. Todavia, é difícil manter um quadro claro do mapa de conceitos quando expressões complexas são usadas.

O sistema AHA oferece ainda inclusão condicional de fragmentos. Uma página pode conter um número arbitrário de fragmentos que são incluídos se uma expressão *booleana* é satisfeita. A ordem dos fragmentos é fixa.

Uma alternativa para o AHA seria a utilização de arquivos separados para cada fragmento e a operação de inclusão no servidor para a montagem das páginas a partir dos fragmentos. Seria fácil colocar em diferentes ordens para diferentes usuários ou incluir fragmentos em várias páginas. A desvantagem seria a sobrecarga quando os fragmentos são muito pequenos.



Mais adiante no item 3.8, apresenta-se diversos exemplos de aplicações desenvolvidas incluindo comentários relativos às arquiteturas utilizadas.

### 3.2.2 O modelo do usuário

O modelo do usuário é fortemente dependente do domínio do conhecimento.

Estudos desenvolvidos na área de tutores inteligentes, principalmente na década de 90, apresentam diversas técnicas de representação do modelo do domínio e do modelo do usuário. Muitas dessas técnicas estão sendo adaptadas para SHA, criando um vínculo entre as tecnologias. O funcionamento de um sistema hipermídia adaptativo prevê a obtenção de informações pessoais dos usuários e suas interações junto ao sistema, via navegador. Com base nessas informações o sistema adapta seus conteúdos e possibilidades de navegação para cada usuário. Em outras palavras, os SHA devem possuir um modelo do usuário e devem ser capazes de adaptar a hipermídia do sistema a esse modelo. A figura 3 abaixo ilustra o modelo de adaptação.

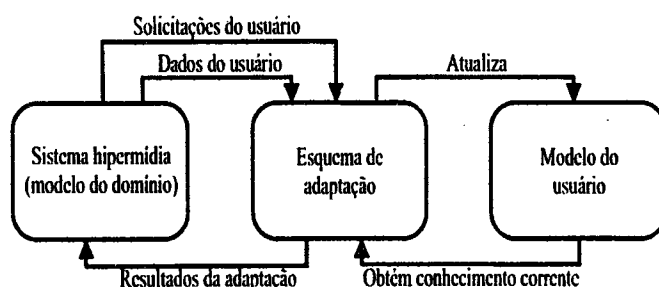


Figura 3: Visão geral de um SHA

Brusilovsky (1996), identificou cinco características associadas aos usuários que são importantes para o sistema: conhecimento, objetivos, preferências, histórico e experiência. Essas características são de alguma forma dinâmicas o que requer um modelo do usuário atualizado continuamente. A seguir comenta-se:

- **O Conhecimento** - o conhecimento do usuário sobre os assuntos do domínio é a principal informação para a adaptação, principalmente em se tratando de sistemas aplicados à educação. Tendo em vista a dinamicidade do conhecimento, o sistema deve ser capaz de controlar dinamicamente as modificações e realimentar o modelo

do usuário. Duas técnicas são normalmente utilizadas para a modelagem do conhecimento do usuário: técnica de sobreposição (HOHL et al., 1996) e técnica de estereótipos (BOYLE; ENCARNACION, 1994).

A técnica de **sobreposição** (*overlay*) tem como base a comparação do atual nível de conhecimento do usuário com o conhecimento de um *expert*. A parte crítica dessa técnica está na estimativa do nível de conhecimento inicial do usuário. A cada conceito do domínio é associado um valor que estima o nível de conhecimento do usuário sobre este conceito. Temos então um par conceito-valor na técnica de sobreposição.

Na técnica de **estereótipos**, para os usuários pertencentes à mesma classe são assumidas as mesmas características. Nessa técnica podem ocorrer problemas se houver dificuldades em enquadrar um usuário num determinado estereótipo ou então se forem adotados estereótipos muito especializados.

- **Objetivos** - tem relação direta com o uso do sistema como ferramenta para, por exemplo, desenvolver atividades de trabalho (em sistemas aplicativos), pesquisar assuntos de interesse, alcançar metas de aprendizagem num sistema educacional. Os objetivos de um usuário variam muito de uma sessão para outra ou mesmo na mesma sessão. Uma maneira interessante para modelar os objetivos correntes do usuário é através de pares objetivo-valor, onde valor seria, por exemplo, a probabilidade que o objetivo correspondente seja o objetivo corrente do usuário (MICCARELLI; SCIARRONE, 1996). Vassileva (1996) propõe a representação de possíveis objetivos do usuário através de uma hierarquia (árvore) de tarefas relacionadas.
- **História** - a história do usuário em tópicos relevantes, não relacionados propriamente com os assuntos do sistema hipermídia, é fator importante na produção do modelo do usuário. Por exemplo, conhecimentos sobre a linguagem Pascal podem ser considerados como experiência relevante para um usuário, num sistema hipermídia de um curso de Java.
- **Experiência** – refere-se à experiência do usuário com sistemas baseados em hipertexto ou hipermídia, adaptativos ou não.
- **Preferências** - os usuários de sistemas de hipermídia podem ter diferentes preferências, por exemplo, fontes, cores etc. Essas características devem ser obtidas

junto aos usuários. Sistemas de hipermídia adaptativa devem ser capazes de oferecer diversas alternativas de preferências possibilitando que os próprios usuários façam as suas escolhas.

### **3.2.3 Modelo de adaptação**

O modelo de adaptação é o motor central que conduz o processo de adaptação num SHA. Ele capta as informações do usuário e suas interações no ambiente hipermídia (que contem o modelo do domínio) e mantém atualizado o modelo do usuário. Em outra instância obtém dados do modelo do usuário, já atualizado e promove a adaptação da interface. Isto pode ser visto melhor na figura 3 já apresentada anteriormente.

A adaptação de conteúdo de informação (adaptação de apresentação de tópicos) e adaptação de links de navegação é normalmente baseada em um conjunto de regras. Essas regras fazem a ligação entre o modelo do usuário, o modelo do domínio e a apresentação a ser gerada. Normalmente são definidas regras gerais, que utilizam variáveis que representam conceitos e relações entre conceitos. Convém lembrar que entre conceitos poderá haver condições de pré-requisitos, que devem ser mapeadas na rede de conhecimentos pelo professor autor. Regras específicas usam conceitos concretos do domínio ao invés de variáveis. Regras gerais são freqüentemente regras do sistema, significando que o autor não precisa especificá-las. Por exemplo, uma instância da regra poderá determinar como o nível de conhecimento de um conceito arbitrário C1 influencia a relevância de outro conceito para o qual C1 é pré-requisito. As regras do autor têm sempre precedência sobre as regras gerais do sistema, no caso de conflitos (DE BRA; HOUBEN; WU, 1999).

Enquanto que regras específicas são tipicamente utilizadas para criar exceções das regras gerais, elas podem ser também usadas para executar algumas adaptações do tipo locais, baseadas em conceitos para os quais o modelo do domínio não provê relacionamento.

## **3.3 Espaço de adaptação**

Segundo Brusilovsky (1996), ao se falar de sistemas adaptativos, surgem as seguintes perguntas: o que pode ser adaptado? Quais as características do sistema que variam de usuário para usuário? Qual é o espaço para possíveis adaptações? Em sistemas adaptativos o espaço

para adaptações é um tanto delimitado, e envolve a adaptação de conteúdo das páginas (adaptação de conteúdo) e adaptação dos links de páginas regulares, índices de páginas e mapas (adaptação nos links ou navegação adaptativa). A figura 4 abaixo ilustra os diferentes espaços de adaptação relativos a hipermídia adaptativa clássica.

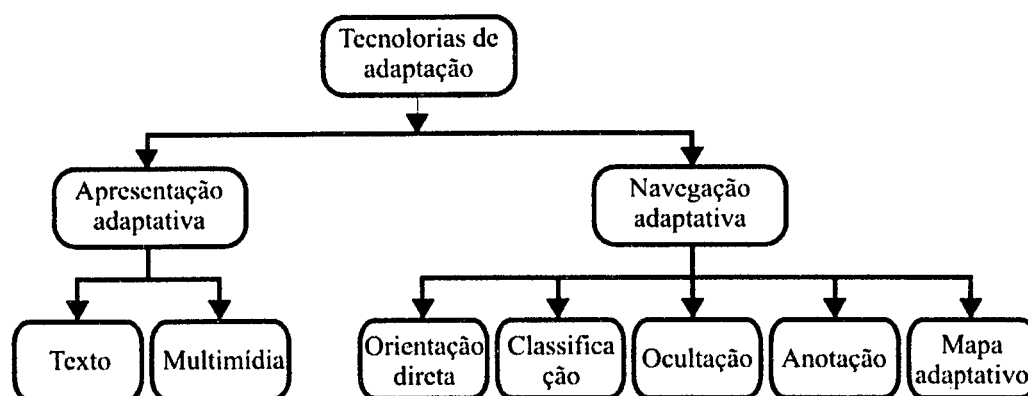


Figura 4. – tecnologias de adaptação em hipermídia adaptativa clássica (BRUSILOVSKY, 1996).

Diversos sistemas baseados em hipermídia adaptativa, surgidos a partir do início dos anos 90, implementaram uma grande variedade de métodos e técnicas de adaptação, algumas das quais serão abordadas a seguir.

### 3.4 Métodos da apresentação adaptativa

Na apresentação adaptativa, procura-se moldar o conteúdo de um nó acessado de acordo com as necessidades do usuário, por exemplo, escondendo informação muito especializada ou inserindo explicações adicionais. Usuários qualificados podem receber informação com maior nível de profundidade enquanto que a um iniciante podem ser oferecidas explicações adicionais para melhor entendimento do assunto tratado.

Brusilovsky (1996) identifica os seguintes métodos para a adaptação de conteúdo:

- Explicação Adicional (EA);
- Explicação Requerida (ER);
- Explicação Comparativa (EC);

- Explicação Variante (EV);
- Classificação de Fragmentos (CF).

O método da **Explicação Adicional (EA)** procura ocultar partes do conteúdo que não são relevantes para o usuário num determinado momento. Pode ocorrer que determinada classe de usuários necessite de informações adicionais, especialmente preparadas, que não são mostradas às outras classes. Em outra situação é necessário ocultar certas partes da informação que não são relevantes para uma determinada classe de usuários. Para exemplificar, esse método é usado nos sistemas MetaDoc (BOYLE; ENCARNACIÓN, 1994) e KN-AHS (KOBISA et al, 1994).

O método da **Explicação Requerida (ER)** é informação adicional sobre conceitos que são pré-requisitos do conceito em estudo. Neste caso, o modelo de adaptação checa os pré-requisitos para entendimento de uma página solicitada pelo usuário e anexa, se necessário, a informação correspondente. Seguindo essa idéia, ao apresentar a explicação de um conceito, o sistema insere a explicação de todos os conceitos requeridos para o seu entendimento. Este método é empregado no sistema C-book (KAY; KUMMERFELD, 1994).

O método da **Explicação Comparativa (EC)** tem como base a similaridade existente entre conceitos. Se um conceito, que está sendo apresentado, é similar a um conceito já conhecido, o usuário recebe uma explicação comparativa realçando as semelhanças e diferenças entre os conceitos. Este método é bastante utilizado no aprendizado de linguagens de programação. Utilizam este método, por exemplo, os sistemas C-book (KAY; KUMMERFELD, 1994), Lisp-Critic (FISHER et al, 1990) e Item-IP (BRUSILOVSKY, 1992).

O método da **Explicação Variante (EV)**, parte do princípio que usuários diferentes podem necessitar informações essencialmente diferentes. Neste método o sistema cataloga diversas variantes para alguns dos conteúdos de uma página e o usuário obtém a apresentação da EV que corresponde ao seu modelo. Este método é empregado em Anatom-Tutor (BEAUMONT, 1994) e Hypadapter (HOHL; BÖCKER; GUNZENHÄUSER, 1996).

O método **Classificação de Fragmentos (CF)** leva em conta o nível de conhecimento e a experiência do usuário ordenando fragmentos de informação sobre o conceito, de modo que a informação mais relevante para o usuário (de acordo com o seu modelo) seja apresentada primeiro. Este método é implementado pelos sistemas Hypadapter (HOHL; BÖCKER; GUNZENHÄUSER, 1996) e EPIAIM (DE ROSIS et al., 1993).

### 3.5 Técnicas da apresentação adaptativa

As seguintes técnicas são utilizadas para a implementação dos métodos de apresentação adaptativa descritos anteriormente (BRUSILOVSKY, 1996):

- Texto condicional (TC);
- Técnica de Stretchtext;
- Técnicas de Fragmento Variante (FV) ou Página Variante (PV);
- Técnica de uso de Frames;
- Técnica Stretchtext com adaptação por meio de Frames.

A técnica de **Texto Condicional (TC)** é uma técnica bastante simples e eficiente que trata de inclusão condicional de fragmentos (textos, imagens etc.). Para exemplificar, divide-se a informação em diversas porções de texto, sendo cada porção associada a uma ou mais condições relacionadas com o conhecimento do usuário. Na navegação o sistema mostrará apenas as porções de texto que tiveram suas condições satisfeitas. Essa técnica é muito flexível e permite implementar todos os métodos de adaptação relacionados acima, à exceção da Classificação de Fragmentos. A técnica **Texto Condicional (TC)** é utilizada nos sistemas Item-IP (BRUSILOVSKY, 1992), C-book (KAY E KUMMERFELD, 1994) e Lisp-Critic (FISHER et al., 1990).

A técnica de *Stretchtext* permite também apresentar ou ocultar condicionalmente porções de texto ou outros tipos de fragmentos de acordo com o conhecimento do usuário. Foi implementada no sistema Metadoc (BOYLE; ENCARNACIÓN, 1994) e posteriormente desenvolvida em KN-AHS (KOBISA et al., 1994). Na técnica de *stretchtext* os *links* podem ser expandidos para seus conteúdos ou concentrados novamente em uma palavra chave. Neste contexto, as informações relevantes para o usuário, num dado momento, estarão expandidas e as não relevantes estarão representadas por uma palavra ou frase.

As técnicas de **Fragmento Variante (FV)** ou **Página Variante (PV)** permitem a implementação do método EV. Na técnica **PV** são construídas duas ou mais páginas alternativas para um mesmo conceito, cada uma adaptada a uma certa classe de usuário. Essa técnica é empregada nos sistemas Anatom-Tutor (BEAUMONT, 1994) e C-Book (KAY; KUMMERFELD, 1994). Uma técnica Similar foi empregada em Epiaim (De ROSIS et al., 1993) para a apresentação de exemplos relativos a um determinado conceito, conforme a experiência ou nível de conhecimento do usuário.

A técnica **FV** é aplicada no caso em que não há correspondência entre uma página e um conceito, ou seja, uma página pode apresentar vários conceitos. Como exemplo cita-se o sistema Anatom-Tutor (BEAUMONT, 1994) no qual cada conceito pode possuir diversos fragmentos variantes e a página é montada com a combinação que melhor se adapta ao conhecimento e experiência do usuário. Na realidade o sistema Anatom-Tutor combina as técnicas **PV** e **FV**, objetivando atender melhor as necessidades dos usuários.

A técnica do uso de **frames** é potencialmente a mais abrangente e é considerada a mais eficiente de todas as técnicas de apresentação adaptativa, tendo sido utilizada, por exemplo, nos sistemas Hypadapter (HOHL; BÖCKER; GUNZENHÄUSER, 1996) e Epiaim (De ROSIS et al., 1993). A informação sobre um determinado conceito é representada sob a forma de um *frame*. Os *slots* do *frame* podem conter diversas **EV** sobre o conceito, links para outros *frames*, exemplos, etc. O *frame*, na sua estrutura permite a catalogação de regras de apresentação para decidir quais os *slots* devem ser apresentados a um certo usuário e em que ordem específica isto deve ocorrer. No sistema Epiaim essas regras são empregadas para selecionar um dentre vários esquemas de apresentação disponíveis (onde cada esquema é um conjunto ordenado de slots) que é usado para a apresentação do conceito. No sistema Hypadapter, as regras são empregadas para calcular a prioridade de apresentação de cada slot. Em seguida o subconjunto de slots com maior prioridade é apresentado ao usuário.

A técnica **stretchtext com a adaptação por meio de frames** é bastante eficiente tendo sido usada no sistema de informações *on-line* Push (HÖÖK et al., 1998). Neste sistema cada página contém diferentes tipos de informações para cada objeto componente, desempenhando papel semelhante executado pelos *slots* de um *frame*. A descrição de um objeto neste sistema, entretanto costuma ser muito grande, ocupando diversas páginas de hipertexto. Através da técnica de **stretchtext** o sistema mostra as porções de informações sobre o objeto corrente que são relevantes para o usuário num dado momento.

A tabela 1 abaixo apresenta a associação dos diferentes métodos e técnicas descritos com os sistemas citados que as implementam.

Técnicas Métodos	Texto Condicional	Stretchtext	Fragmentos Variantes	Páginas Variantes	Frames
Explicações Adicionais, Requeridas e Comparativas.	C-Book Item/Ip Lisp-Critic	Metadoc Kn-Ahs Push			Epiaim Push
Explicações Variantes	C-Book		Anatom-Tutor Lisp-Critic	Anatom-Tutor C-Book Epiaim	Hypadapter
Classificação					Epiaim Hypadapter

Tabela 1 - Métodos e Técnicas de Apresentação Adaptativa em SAH (BRUSILOVSKY, 1996)

### 3.6 Métodos da navegação adaptativa

A navegação adaptativa tem como objetivo personalizar as possibilidades de navegação do usuário no sistema de hipermídia, sendo a manipulação de links a base do processo.

Os links são classificados segundo quatro diferentes classes (BRUSILOVSKY, 1996):

- **Links locais, não contextuais:** são *links* que não tem relação com o conteúdo do nó em que se encontram e são normalmente apresentados como botões, uma lista ou um menu *pop-up*;
- **Links contextuais ou “hipertexto verdadeiro”:** são links relacionados diretamente ao contexto do conteúdo do nó, representados normalmente no texto por certas palavras ou frases ou por zonas especialmente delimitadas de uma imagem. Estes *links* podem ser anotados, mas não podem ser classificados nem totalmente ocultos;
- **Links para índices e tabelas de conteúdos:** um nó com índices ou tabelas de conteúdo pode ser visto como um tipo especial de nó que só contém links. Tais *links* são assumidos como não-contextuais, a menos que sejam implementados sob a forma de uma imagem;
- **Links para mapas locais e globais:** mapas são representações gráficas de um hiperespaço ou área local de um hiperespaço como uma rede de nós conectados por setas. O usuário pode navegar diretamente sobre todos os nós visíveis no mapa.



Segundo Brusilovsky (1996), os principais métodos de suporte à navegação adaptativa são:

- Condução Global;
- Condução Local;
- Suporte à Orientação Local (Conhecimento);
- Suporte à Orientação Local (Objetivos);
- Suporte à Orientação Global.

A **Condução Global (CG)** pode ser oferecida quando o usuário possui algum objetivo definido em relação à informação desejada. Condução global significa que o sistema sugere caminhos de navegação numa escala global para o usuário atingir o seu objetivo. No caso de sistemas hipermídia voltados para a educação, quando um usuário quiser estudar determinado tópico, o sistema pode sugerir um conjunto de páginas a serem lidas, com uma indicação da ordem de leitura. Através da **CG** o usuário é levado a encontrar o caminho mais curto para a informação desejada. A condução global é de fundamental importância em sistemas de recuperação de informações hipermídia, em sistemas de informações de um modo geral, bem como em sistemas de ajuda on-line. A **CG** é empregada no sistema WebWatcher (ARMSTRONG; FREITAG, 1995). Um método complementar é a classificação dos links que serão sugeridos ao usuário em ordem decrescente da relevância que possuem para os objetivos do usuário. Os sistemas Adaptive HyperMan (MATHÉ; CHEN, 1996) e Hyperflex (KAPLAN et al., 1993) implementam este método.

A **Condução Local (CL)** sugere ao usuário o próximo passo a tomar, oferecendo os *links* relevantes que podem ser usados a partir do nó corrente, como por exemplo, o botão “Próximo” ou “Continue”. O objetivo da **CL** é mais modesto do que a condução global (**CG**), mas existe semelhança entre os métodos. Como exemplo, os sistemas Adaptive HyperMan e Hyperflex utilizam um método para condução local que consiste em classificar os links de acordo com as preferências do usuário.

O suporte à **Orientação Local (OL)** tem como objetivo fazer o usuário entender o que está em sua volta, e a sua posição no hipertexto em termos locais. Para alcançar este objetivo é fornecida informação adicional sobre os nós que podem ser acessados a partir do nó corrente, ou então são diminuídas as oportunidades de navegação para evitar a **sobrecarga cognitiva**, fato este que pode ocorrer em função do excesso de informação. A técnica de

**ocultação** é normalmente utilizada, removendo-se toda a informação que não é relevante para os objetivos do usuário num dado momento. Dentro dessa técnica há vários métodos de implementação, sendo o mais simples mostrar somente os *links* relevantes para as preferências dos usuários (WATERWORTH, 1996). Entretanto o método de implementação mais empregado é o que considera o **objetivo corrente** do usuário, utilizado nos sistemas HyPlan (GRUNST, 93), Hynecosum (VASSILEVA, 1996) e Push (HOOK, 1998). Um outro método, adotado no sistema HyperTutor (PÉREZ et al., 1995), consiste em mostrar mais links para os usuários com maior experiência. Os usuários novatos vêem somente um pequeno número de links, que vai crescendo na medida em que sua experiência aumenta.

Em ambientes educacionais dois métodos de **OL** são os mais empregados: o primeiro, consiste na ocultação dos nós que o usuário não está preparado para aprender, empregado, por exemplo, nos sistemas Isis-Tutor (BRUSILOVSKY; CALVI, 1994), HyperTutor (PÉREZ et al., 1995), e Hypadapter (HOH et al., 1996). O segundo, específico para sistemas educacionais adaptativos, consiste em esconder os links para nós que pertençam a objetivos educacionais de outras lições e estejam fora dos objetivos da presente lição. Isso é feito nos sistemas Isis-Tutor e Sypros (GONSCHOREK; HERZOG, 1995).

A **Orientação Global (OG)** tem como objetivo fazer o usuário entender a estrutura de todo o hiperespaço e a sua absoluta posição no mesmo. Para alcançar este objetivo os sistemas de hipermídia adaptativa (SHA) oferecem como suporte, principalmente, técnicas de ocultação e anotação, tendo como base o modelo do usuário, independentemente da sua posição no hiperespaço. No caso da **OG**, o método mais amplamente utilizado é aumentar gradualmente o número de links visíveis, na medida em que cresce a experiência do usuário no hiperespaço considerado. Isto ocorre nos sistemas HyperTutor e Hynecosum (VASSILEVA, 1996).

### 3.7 Técnicas da navegação adaptativa

Brusilovsky (1997) identifica as principais formas de se tratar a navegação adaptativa como: orientação direta, classificação de links, anotação de links, ocultação de links, remoção de links e mapas adaptativos.

A técnica de **Orientação Direta (OD)** é bastante utilizada e de fácil implementação. Permite ao usuário visualizar qual o melhor nó a ser visitado a seguir, considerando seus

objetivos e outros parâmetros do modelo do usuário. Para tanto o sistema pode destacar visualmente o *link* para o melhor nó, como é feito no WebWatcher (ARMSTRONG et al., 1995) ou apresentar um *link* dinâmico adicional normalmente denominado *next* que é conectado ao melhor nó selecionado, como no sistema Isis-Tutor (BRUSILOVSKY; PESIN, 1994). OD é uma escolha óbvia para implementar o método da condução local. No entanto é aconselhável que essa técnica seja usada com outras técnicas complementares, pois normalmente ela não oferece suporte quando o usuário não segue a sugestão do sistema. A OD pode ser aplicada em todos os métodos de navegação adaptativa citados anteriormente.

A técnica de **Classificação de Links** consiste em classificar todos os links de uma determinada página e apresenta-los em ordem decrescente de relevância, tendo em vista o modelo do usuário. Essa técnica é muito utilizada em sistemas de recuperação de informações (que classificam links para páginas de acordo com critérios pessoais relevantes), sistemas de documentação *on-line* como também em aplicações de hipermídia educacional nas quais os usuários selecionam objetivos de aprendizado e o sistema hipermídia gera uma lista de páginas numa ordem que considera os pré-requisitos de estudo. Quanto à aplicabilidade, verifica-se que essa técnica não pode ser aplicada para links contextuais e mapas. Aplica-se facilmente em links não contextuais como pode ser visto em Kaplan et al. (1993), reduzindo significativamente o tempo de navegação em sistemas de recuperação de informação. No caso de tabelas de conteúdo e índices a aplicação é muito difícil.

A técnica de **Anotação de Links** coloca informação adicional nos links através de algum tipo de anotação ou comentário. A informação adicional pode estar na forma de texto ou como indicadores visuais, tais como ícones especiais, cores, diferentes fontes de caracteres ou fontes com tamanhos diferentes. Em se tratando de SHA, espera-se que a anotação seja dinâmica, orientada ao modelo do usuário. O sistema Interbook (BRUSILOVSKY, 1998) utiliza bolas coloridas e setas para indicar o estado dos links. A técnica da anotação mantém uma ordenação estável dos links e em geral é mais poderosa que do que a ocultação, pois essa última oferece dois estados possíveis: relevante (visível), não relevante (oculto). Além disso a anotação pode simular a ocultação, obscurecendo itens não relevantes. Itens obscurecidos se manteriam visíveis e operacionais, se necessário. A anotação de links informa ao usuário sobre o estado corrente das páginas apontadas. Adicionalmente, podem ser apontadas três possibilidades para a anotação de links:

- a anotação pode indicar a relevância do link. Cores podem ser usadas para distinguir para um dado instante, altamente relevante, um pouco relevante e não relevante;

- a anotação pode indicar se o usuário já conhece o conceito descrito na página apontada pelo link. Vários graus de conhecimento podem ser distinguidos: não conhecido, conhecido e bem conhecido;
- a anotação pode indicar se o usuário está habilitado para entender a informação contida na página destino: pronta para ser lida e não pronta para ser lida.

Quanto á aplicabilidade, a anotação pode ser usada para todos os tipos de links.

A técnica de **ocultação de links** é um caso especial da técnica de anotação e é muito utilizada na navegação adaptativa. Links que são considerados não relevantes ou não desejáveis no momento, são apresentados como texto normal (não colorido). Um determinado nó é considerado relevante se está relacionado com os objetivos do usuário ou se o mesmo está apto para acessá-lo. Para controle da ocultação de links pode ser utilizada a técnica baseada num conjunto de regras pedagógicas que decidem quais conceitos e nós estarão ou não visíveis num dado momento. Essas regras levam em conta as relações entre conceitos e o nível de conhecimento já alcançado pelo usuário. Se o sistema decide que um determinado nó não deverá estar visível então todos os links contextuais para esse nó estarão ocultos, dissolvidos como texto comum. A ocultação pode ser aplicada a todos os tipos de links.

A técnica de **remoção de links** é uma técnica utilizada normalmente em listas de links, simplesmente removendo os links não relevantes. Essa técnica, de um modo geral, é utilizada junto com a classificação de links; feita a classificação, somente são mostrados os links relevantes e os demais são removidos.

A técnica dos **Mapas Adaptativos (MA)** envolve diversas formas de adaptação de mapas de hipermídia global e local apresentados ao usuário. As técnicas de orientação direta, ocultação e anotação podem também ser usadas na adaptação de mapas hipermídia, entretanto o emprego de tais técnicas não modifica a forma ou a estrutura dos mapas. A pesquisa realizada na área da interação homem-máquina oferece diversas técnicas para a adaptação da forma e estrutura de diversos tipos de redes, incluindo mapas hipermídia (MUKHERJEA; FOLEY, 1995). Tais técnicas entretanto, não foram ainda aplicadas em sistemas de HA. A única exceção conhecida é o sistema Hypercase (MICARELLI; SCIARRONE, 1996).

A tabela 2, fonte Brusilovsky (1996), apresenta uma visão da aplicação de métodos e técnicas em diversos sistemas de hipermídia adaptativa. As diversas técnicas podem ser combinadas, ou seja, elas não são mutuamente exclusivas nem contraditórias. Por exemplo o sistema Isis-Tutor (BRUSILOVSKY; PESIN, 1994) emprega orientação direta, ocultação e anotação e o Hypadapter (HOHL et al, 1996) usa classificação e ocultação. Em particular a

orientação direta pode ser naturalmente empregada em combinação com qualquer outra técnica.

<b>Técnicas Métodos</b>	<b>Orientação Direta</b>	<b>Classificação</b>	<b>Ocultação</b>	<b>Anotação</b>	<b>Mapas Adaptativos</b>
<b>Condução Global</b>	WebWatche Item-Ip Hyperflex	HyperMan Hyperflex			
<b>Condução Local</b>	HyperTutor	HyperMan Hypadapter Hyperflex	Hypadapter Push	Isis-Tutor Elm-Art	Hypercase
<b>Orientação Local - Conhecimento</b>		Hypadapter	HyperTutor Hypadapter Isis-Tutor	Isis-Tutor Elm-Art Item-Pg	
<b>Orientação Local - Objetivos</b>			Hynecosum HyPlan ISIS-Tutor PUSH Sypros	Isis-Tutor Elm-Art	Hypercase
<b>Orientação Global</b>			Hynecosum HyperTutor Isis-Tutor Sypros	Isis-Tutor Elm-Art Item-Pg	Hypercase

Tabela 2 - Métodos e Técnicas de Navegação Adaptativa em Sistemas de Hipermídia Adaptativa (BRUSILOVSKY, 1996).

### 3.8 Exemplos de sistemas de hipermídia adaptativa aplicados na educação.

A seguir comenta-se alguns sistemas de hipermídia adaptativa aplicados na educação.

#### 3.8.1 Elm-Art

Segundo Henze (2000), o sistema Elm-Art (BRUSILOVSKY; SCHWARZ; WEBER, 1996) e seus sucessores Elm-Art II (WEBER; SPECHT, 1997) e Interbook (BRUSILOVSKY; SCHWARZ; WEBER, 1996a), foram os primeiros sistemas do tipo SHA a serem colocados na WEB. São baseados no sistema Elm-PE construído para funcionar de maneira isolada, aplicado num curso introdutório de LISP.

No sistema Elm-Art é utilizado um modelo episódico do aluno para o diagnóstico de soluções completas e incompletas do problema. O modelo episódico armazena o conhecimento sobre o aluno na forma de uma coleção de episódios. Estes episódios são comparáveis aos casos na aprendizagem baseada em casos. Para construir o modelo do aluno, o código de programação produzido por ele é analisado de acordo com o conhecimento do domínio e de uma descrição da tarefa. O resultado cognitivo deste diagnóstico resulta em uma árvore de derivação dos conceitos e das regras que o aluno pôde ter usado. Em Elm-Art, os conceitos são relacionados por seus pré-requisitos e efeitos. Assim, uma rede conceitual é construída. As observações sobre o usuário são feitas monitorando as páginas visitadas: o conceito que corresponde a uma página visitada é marcado na rede conceitual como sabido. Para anotar os links, os autores usam a metáfora da luz de tráfego. Uma esfera vermelha indica as páginas que contêm a informação para as quais o aluno não está preparado, uma esfera verde indica os links sugeridos. Elm-Art contém também os exemplos interativos, que podem ser traduzidos com um compilador do Lisp através da Web.

Elm-Art II foi desenvolvido para traduzir livros normais em livros eletrônicos. Elm-Art II melhora a representação do conhecimento de Elm-Art. A rede conceitual é organizada hierarquicamente em lições, seções, subseções e em páginas terminais. Cada unidade na rede conceitual tem um espaço com o texto para a página e a informação para relacionar essa página a outras unidades. Os Espaços estáticos contêm conceitos pré-requisitos, conceitos relacionados, e efeitos. As páginas terminais contêm espaço para testes. As páginas de problemas têm um espaço específico para armazenar uma descrição do problema de programação. O modelo individual do aluno armazena páginas visitadas, testes resolvidos, e problemas de programação resolvidos marcando os conceitos correspondentes no modelo conceitual como "sabido". A orientação direta é fornecida pela tecla "melhor próxima". A ajuda é proposta procurando mostrar o exemplo mais relevante do histórico individual da aprendizagem, baseado em um diagnóstico do código de programação das soluções dos estudantes. O sistema pode fazer inferências sobre o conhecimento dos usuários baseado nos

conceitos marcados no modelo conceitual. Todos os pré-requisitos dos conceitos sabidos são marcados também recursivamente como sabidos.

### 3.8.2 Interbook

É uma ferramenta que facilita a criação de livros eletrônicos adaptativos (BRUSILOVSKY; SCHWARZ; WEBER, 1996a). **Interbook** utiliza um processo de transformação de um livro eletrônico em um livro adaptativo. Este processo consiste basicamente em estruturar o conteúdo especificando a relação entre um conceito e seus pré-requisitos e resultados. O Interbook utiliza o MS\_Word não só para a edição e formatação de documentos, mas também para definir as relações entre os nós, o que é uma tarefa árdua quando o usuário não tem preparação para trabalhar neste nível.

Diversas etapas tais como criar uma lista de conceitos do domínio, estruturar e anotar as páginas, definir pré-requisitos, tradução para o HTML e tradução da informação para uma estrutura do Lisp, têm que ser executadas para obter um Interbook (livro eletrônico). Interbook usa um domínio e um modelo do usuário. O glossário e o livro texto são baseados no modelo do domínio. O glossário é uma rede visualizada do domínio. A estrutura do glossário assemelha-se à estrutura didática do conhecimento do domínio. Cada item do glossário corresponde a um dos conceitos do domínio. Adicionalmente, cada entrada do glossário fornece os links a todas as seções do livro que usam este conceito.

Cada unidade do livro texto é posicionada com alguns conceitos do modelo do domínio. Estes conceitos têm diferentes papéis. Alguns dos conceitos descrevem o conhecimento que um usuário tem após ter lido a página, e outros o pré-requisito que é necessário para ler a página.

Interbook suporta a anotação adaptativa dos links usando a metáfora da luz do tráfego. Executa uma ajuda baseada em pré-requisitos, apresentando uma lista anotada das páginas que contêm a informação pré-requisito.

### 3.8.3 Tangow

O Sistema Tangow é uma ferramenta orientada a objetos para o desenvolvimento de cursos na Internet (CARRO et al., 2000). Possui a habilidade de conduzir o estudante durante o processo de ensino e aprendizagem, tendo como base o seu perfil e ações desenvolvidas no

curso. O currículo seqüencial é gerado dinamicamente de tal forma que os conceitos podem ser vistos de formas diferentes de acordo com as ações do estudante na sua interação no curso.

No sistema Tangow os cursos são descritos em tarefas docentes e regras. As tarefas são unidades que o projetista do curso considerou básicas e que devem ser aprendidas pelo estudante. As tarefas podem ser teóricas, praticas ou exemplos, de acordo com seus conteúdos.

Cada tarefa pode ser atômica ou composta quando se subdivide em outras tarefas. Cada tarefa pode ter associada a uma lista de elementos multimídia: textos, imagens, sons, vídeos, sons, animações etc.

No sistema as regras são utilizadas para descrever a forma em que as tarefas compostas se dividem em sub-tarefas podendo existir várias regras de decomposição para uma mesma tarefa composta. Uma das características que determina cada regra é a seqüência de ativação das subtarefas em que se decompõe a tarefa principal. Pode ser definido que todas as tarefas devem se realizadas na ordem indicada (seqüência AND), em qualquer ordem (seqüência any), ou que se realizem apenas algumas delas (seqüências OR e XOR). Também pode ser especificada na descrição das regras a condição em que as tarefas deverão acontecer para a sua aplicação. Essas condições podem depender do resultado da informação relativa a forma como o estudante tem realizado as tarefas previas, das características do estudante e também da estratégia de aprendizagem.

#### **3.8.4 PT**

PT é um ambiente de suporte para ensino da linguagem de programação C (KAY; KUMMERFELD, 1997). O material didático está organizado como um livro convencional e considera que o estudante já conhece a linguagem Pascal. São utilizados estereótipos do usuário alvo (programadores do Pascal) junto com um modelo individual. No processo de adaptação o PT usa as similaridades entre a linguagem Pascal e a linguagem C para apresentar a informação aos usuários.

#### **3.8.5 AHA**



No sistema AHA (DE BRA; CALVI, 1999), desenvolvido na Universidade de Eindhoven, a adaptatividade é obtida mediante a inclusão de sentenças condicionais em documentos HTML, como comentários. Essas sentenças agem como filtros para decidir em tempo de execução quais os fragmentos de HTML mostrar para os estudantes. O modelo do usuário é mantido através de parâmetros *true* ou *false*, para indicar se o estudante conhece determinado conceito. Estes parâmetros são usados durante o processo de avaliação de sentenças condicionais, e atualizados por meio de outras sentenças que são avaliadas em tempo de execução. A ferramenta é totalmente voltada para a WEB e os autores precisam colocar em seus documentos HTML sentenças condicionais que serão avaliadas na execução. A desvantagem nesse processo de inserir páginas nos documentos HTML está na dificuldade de se criar e manter um grande número de páginas com diferentes situações. Outra questão que se coloca está na dificuldade de reutilização do material didático em cursos similares, pois os documentos HTML gerados contêm elementos próprios para controle da adaptatividade.

### 3.8.6 Olae e Pola

**Olae** (VANLEHN, 1998) foi construído para avaliar o nível de conhecimento de usuários em um domínio da física. O sistema coleta dados do usuário resolvendo problemas da física elementar através de uma interface apropriada. A partir dos dados coletados o sistema utiliza o conceito de redes bayesianas para determinar qual o nível de conhecimento do usuário num dado domínio de conhecimento. Dessa maneira é elaborado um diagnóstico retrospectivo, chamado traço do conhecimento. **Pola** (CONATI; VANLEHN, 1997) foi projetado para executar o traço do modelo. Ele pode ser invocado repetidamente durante o processo de resolução do problema fazendo previsões sobre as soluções que o usuário está seguindo. Pola constrói a rede bayesiana subjacente incremental adicionando nós cada vez que o estudante executa uma ação (HENZE, 2000).

### 3.8.7 EPI-UMOD

No sistema EPI-UMOD (ROSIS et al., 1992) o modelo do usuário é baseado em estereótipos, utilizando o conceito de redes bayesianas. É construída uma rede bayesiana separada para cada estereótipo em que as dependências condicionais especiais entre itens do

conhecimento são executadas. Cada atributo no estereótipo representa uma indicação que o usuário conhece um determinado conceito particular.

### **3.8.8 KBS-Hiperbook**

KBS-Hyperbook (HENZE, 2000) é um sistema com arquitetura aberta e funciona na Internet. É voltado para a aprendizagem baseada em projetos. O KBS usa a metáfora da luz do tráfego para a anotação adaptável. Um algoritmo de sequenciamento de páginas gera seqüências da leitura de acordo com os objetivos e o conhecimento do usuário. Para ajudar o usuário a achar seu caminho o sistema gera o próximo passo da aprendizagem comparando seu estado atual de conhecimento com o conhecimento que deve ter após terminar o livro. Usuários podem definir seus próprios objetivos da aprendizagem ou podem pedir o objetivo seguinte ao sistema. Para cada um desses objetivos, é gerada uma seqüência da leitura que contém o conhecimento necessário (pré-requisitos e o conhecimento atual). Além disso, os projetos apropriados são selecionados e é apresentado um índice de informações que contem documentos contidos no próprio Hiperbook e utros documentos localizados na Internet. A seleção de projetos apropriados é baseada em algoritmos que refletem ambos, o pré-requisito necessário para executar o projeto, e quanto o projeto combina com o objetivo real da aprendizagem do usuário. O KBS adapta-se também às velocidades diferentes da aprendizagem dos usuários, suportando aprendizagem orientada a objetivos. Os usuários podem dizer o quanto e o que aprender em seguida. Se o sistema observar que um usuário domina bem um projeto avançado, então atualiza sua estimativa sobre esse usuário com relação aos tópicos relacionados bem como os pré-requisitos do conhecimento. Assim o usuário pode avançar em tópicos mais avançados. Se o sistema observar que um usuário não é completamente familiar com algum tópico, propõe os exemplos ou projetos similares, que contêm uma quantidade pequena de informação nova. A técnica de implementação usada no KBS é uma rede bayesiana que opere sobre o modelo completo do domínio da aplicação. Sempre que as observações sobre um usuário particular são feitas, são propagadas através da rede inteira. Um foco do KBS é a extensibilidade do sistema com relação à Internet. Para criar sistemas hipermídia adaptáveis e abertos, a utilização da abordagem de indexação no KBS permite tratar cada unidade de informação com igualdade, independente de sua origem.

Assim, as páginas do HTML do World Wide Web podem ser integradas da mesma maneira que os originais armazenados na biblioteca do Hyperbook.

### 3.9 Conclusão do capítulo

Os SHA constituem uma área de pesquisa recente. Partindo do conhecimento do perfil dos usuários, são capazes de se adaptar às suas necessidades, objetivos e nível de conhecimento, utilizando normalmente recursos e técnicas da área de inteligência artificial.

Estes sistemas reforçam a hipermídia clássica introduzindo um componente inteligente que oferece suporte ao usuário no trabalho com hiperdocumentos. O componente inteligente pode adaptar o conteúdo de uma página ou sugerir o nó mais relevante para a navegação, conforme o nível de conhecimento do usuário. Dessa forma, os SHA procuram evitar os problemas clássicos da hipermídia, a saber, desorientação, sobrecarga cognitiva e sobrecarga de informações.

O aspecto crítico que merecem atenção especial no desenvolvimento de SHA tem relação com a estratégia a ser utilizada para a modelagem do domínio do conhecimento, para a modelagem do usuário e para o processo de autoria. Também é muito importante o projeto da interface de comunicação com o usuário que deve considerar aspectos pedagógicos relacionados com o ambiente.

Neste contexto, um SHA, para cumprir a sua missão de adaptabilidade, precisa aferir dinamicamente, da melhor maneira possível, o que o usuário sabe em relação aos diversos conceitos constantes no domínio do conhecimento e/ou aferir o alcance de objetivos de aprendizagem previamente definidos. Daí percebe-se a relação estreita que deve haver entre o modelo do usuário e o domínio do conhecimento.

Por outro lado o processo de autoria para cursos na Web é crítico em função da dinamicidade e diversidade de cursos e de autores com diferentes perfis. O trabalho do professor autor de cursos é pesado por natureza e deve ser facilitado, evitando-se interfaces baseadas em estruturas de domínio complexas.

Observa-se ainda:

- estruturas de informação e mecanismos de adaptação complexos podem gerar sobrecarga de processamento em servidores e dificuldades de comunicação;

- os SHA devem ser estruturados com a possibilidade de reutilização de material instrucional em diversos cursos, evitando-se redundância na criação de hiperdocumentos (pois documentos podem ser criados sem links embutidos). Uma consequência deste fato é que o custo de produção de material didático fica mais baixo em função da reutilização.
- na questão do modelo do usuário, segundo Brusilovsky (1996), a modelagem automática pelo sistema sem a interferência do usuário não é confiável. Podem ocorrer erros na dedução do modelo do usuário bem como na execução da função de adaptação;
- os SHA são softwares complexos que requerem flexibilidade no controle dos elementos componentes, modularidade, expansibilidade e adaptabilidade em ambientes computacionais distribuídos;
- tendo em vista os objetivos de um SHA é de vital importância intensificar aspectos pedagógicos no projeto da interface com o usuário. Para tanto é preciso utilizar ferramentas de software que resultem em flexibilidade e facilidades de uso para os usuários.

## 4 UTILIZANDO CONCEITOS DE LINGUAGENS FORMAIS EM SHA

### 4.1 Introdução

Ambientes de aprendizagem na Web precisam oferecer maior flexibilidade para a inclusão e manutenção de material instrucional numa base de hiperdocumentos, dando suporte às necessidades cada vez maiores do ensino a distância. Além disso, no contexto deste trabalho, há que se pensar em estruturas capazes também de atender algumas necessidades do processo de adaptação, requeridas em sistemas de hipermídia adaptativa.

Uma característica desejável, apontada por Maurer (1997), é que a criação do material instrucional hipermídia possa ser feita de forma mais independente possível da estrutura de controle de navegação.

É de extrema importância que entendamos que na WWW temos que trocar links por estrutura [...] links embutidos, usados para objetivos estruturais, não devem ser usados em sofisticados ambientes educacionais multimídia em rede [...].

Para tanto, num ambiente de aprendizagem na Web, o material didático seria fragmentado em unidades instrucionais que, transformadas em páginas HTML, seriam colocadas num repositório de uso geral. A ligação entre as diversas unidades seria feita através de links estruturados como componentes externos aos documentos HTML, evitando-se o uso de links de navegação internos. Essa arquitetura permitiria a utilização das unidades instrucionais em diferentes cursos, ou num mesmo curso através da programação de seqüências de estudo com objetivos e enfoques diferenciados.

A separação do material instrucional do controle de navegação é importante para o processo de adaptação em sistemas de hipermídia adaptativa, nos quais é preciso flexibilidade na combinação das unidades instrucionais fragmentadas de acordo com as necessidades específicas de um aluno. Segundo Moreau (1998), para um conjunto de documentos e *links* simples, uma máquina de hipertexto com *links* separados possui a mesma expressividade de

uma máquina com *links* embutidos, ou seja, utilizando *links* externos pode-se obter a mesma estrutura de conexões obtidas com links embutidos.

Neste capítulo serão apresentados e discutidos conceitos de linguagens formais relacionados com a utilização de autômatos finitos para a modelagem e construção da base de hiperdocumentos de sistemas de hipermídia adaptativa.

## 4.2 Autômatos e hipertextos como autômatos – visão teórica

Segundo Machado et al. (1999), uma alternativa para a modelagem e construção da base de hiperdocumentos de cursos na Web é utilizar uma arquitetura baseada num grafo do tipo autômato finito com saída, Máquina de Moore ou Máquina de Mealy. A idéia básica nesta arquitetura é mapear as funções básicas de um hipertexto através de um autômato finito, tendo como resultado final uma estrutura de páginas e *links* num *site* da Web. Nessas máquinas é possível incluir ações de ordem semântica relacionadas com transições ou estados (funções de saída), através das quais são construídas as páginas de saída que são mostradas ao usuário. Uma das vantagens dessa arquitetura é de permitir a utilização de *links* como componentes externos aos documentos *HTML*, evitando-se assim o uso de *links* embutidos. Em consequência é atendido o requisito de independência do material didático do esquema de navegação. Nesse enfoque, um curso na Web pode ser visto sob dois aspectos:

- **Máquina de Moore (HOPCROFT; ULMANN, 1979):** É um autômato finito com saídas associadas aos estados. É representada por uma 7-upla  $M = (\Sigma, Q, \delta, q_0, F, \Delta, \delta_s)$ , onde:
  - $\Sigma$ : alfabeto de símbolos de entrada;
  - $Q$ : conjunto de estados possíveis do autômato, o qual é finito;
  - $\delta: Q \times \Sigma \rightarrow Q$ : função programa ou função de transição, a qual é uma função parcial que, dependendo do estado corrente e do símbolo lido, determina o novo estado do autômato;
  - $q_0$ : estado inicial tal que  $q_0$  é elemento de  $Q$ ;
  - $F$ : conjunto de estados finais tal que  $F$  está contido em  $Q$ ;
  - $\Delta$ : alfabeto de símbolos de saída;
  - $\delta_s: Q \rightarrow \Delta^*$ : função de saída, a qual é uma função total que determina a geração de uma palavra de saída para cada estado.

- **Máquina de Mealy** (HOPCROFT, ULMANN, 1979): é um autômato finito modificado de forma a gerar uma palavra de saída para cada transição. É representado por uma 6-upla  $M = (\Sigma, Q, \delta, q_0, F, \Delta)$ , onde:

$\Sigma$ : alfabeto de símbolos de entrada;

$Q$ : conjunto de estados possíveis do autômato, o qual é finito;

$\delta: Q \times \Sigma \rightarrow Q \times \Delta^*$ : função programa ou função de transição, a qual é uma função parcial que, dependendo do estado corrente e do símbolo lido, determina o novo estado do autômato e a saída gerada;

$q_0$ : estado inicial tal que  $q_0$  é elemento de  $Q$ ;

$F$ : conjunto de estados finais tal que  $F$  está contido em  $Q$ ;

$\Delta$ : alfabeto de símbolos de saída;

**Exemplo:** na figura 5 pode-se ver uma aplicação de autômatos como grafo de controle de um conjunto de hiperdocumentos de um segmento de um curso (MACHADO et al., 1999).

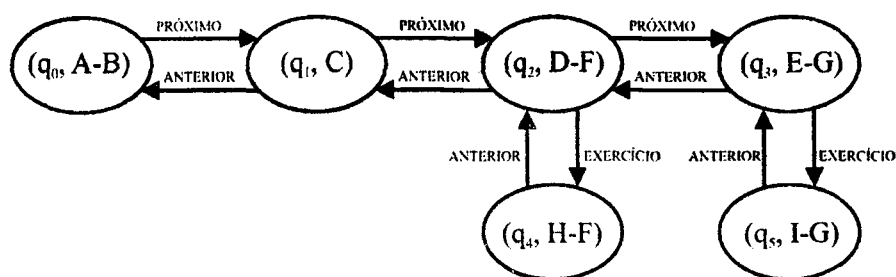


Figura 5 – Modelo de segmento de um curso – máquina de Moore ( MACHADO et al, 1999).

No exemplo da figura 5 tem-se:

estado  $q_0$ : A (definição de autômato finito); B (exemplo de autômato finito);

estado  $q_1$ : C (definição de autômato de saída);

estado  $q_2$ : D (máquina de Mealy); F (exemplo de Mealy);

estado  $q_3$ : E (máquina de Moore); G (exemplo de Moore);

estado  $q_4$ : H (exercícios de Mealy); F (exemplo de Mealy);

estado  $q_5$ : I (exercícios de Moore); G (exemplo de Moore).

Âncoras: próximo, anterior e exercício.

Para o controle da base de hiperdocumentos são feitas as seguintes associações:

- **O alfabeto de entrada  $\Sigma$**  é constituído de nomes que identificam *links* de navegação no hipertexto do curso;
- **O estado inicial  $q_0$**  representa o ponto inicial de entrada no hipertexto (homepage);
- **As saídas do alfabeto  $\Delta$**  estão associadas a unidades de informação (conteúdo de um curso) constituídas por páginas WEB;
- **As palavras de saída  $\Delta^*$**  podem ser vistas como páginas concatenadas como uma única página Web no navegador do usuário.
- **As transições** definidas na função programa funcionam como ligações lógicas (relacionamentos) entre os conteúdos e definem os possíveis links HTML a serem selecionados pelos alunos durante a navegação no hipertexto.
- **Um aspecto importante** do uso de autômatos em sistemas de hipermídia está na possibilidade de composição sobre nós/links (estados e transições) bem como sobre os autômatos. A estrutura formal que permite tal operação de alto nível encontra-se na Teoria das categorias (MACHADO et al., 1999).

Os autores Machado et al (1999) colocam diversas vantagens do uso de autômatos com saída em sistemas de hipermídia (páginas estáticas). Segundo os autores, a independência do material didático do esquema de navegação permite, por exemplo:

- reutilização do material instrucional em diversos cursos;
- construção de seqüências de material instrucional com enfoques diferenciados;
- facilidades na implementação e manutenção das estruturas;
- maior controle dos links que ficam armazenados na estrutura dos autômatos.

**Os trabalhos de Pagano** (apud ALMEIDA, 1999), apresentam uma conceituação teórica e abrangente de **hipertexto**, que será vista a seguir.

Um modelo formal de **hipertexto** ou de **hipermídia** pode ser fundamentado na teoria dos autômatos, incluindo aspectos declarativos (por exemplo, diagramas de transição) e aspectos dinâmicos traduzidos pela evolução do tempo do autômato em resposta a qualquer seqüência de entrada de dados.

Um **autômato** pode ser visto como uma particularização de um sistema dinâmico, entendendo-se o rótulo “dinâmico” como “causal”: as entradas passadas influenciam o futuro, mas o contrário não é verdadeiro. Descreve-se um sistema dinâmico como se estivesse



descrevendo o mecanismo de como ele trabalha, especificando como o conjunto de estados varia com o tempo.

Um **autômato**, em termos abstratos, é descrito como uma sêxtupla (ALMEIDA, 1999):

$A = \{U, Y, X, x_0, \lambda, \eta\}$  onde:

$U$  é o conjunto finito de entradas (excitações);

$Y$  é um conjunto finito de saídas (respostas);

$X$  é um conjunto de estados ou espaço de estado,  $x_0 \in X$  é o estado inicial;

$\lambda: U \times X \rightarrow X$  é a função de próximo estado ou função de transição;

$\eta: U \times X \rightarrow Y$  é a função de próxima saída.

Um autômato é um sistema dinâmico invariante e discreto no tempo. Quando o espaço de estado é um conjunto finito, o autômato é chamado de **autômato finito**. Então a sêxtupla formal é interpretada como sendo uma descrição matemática de uma máquina a qual, se no tempo  $t_0$ , estiver no estado  $x_0$  e receber um segmento de entrada  $u$ , do tempo  $t_0$  ao tempo  $t$ , estará no tempo  $t$  no estado  $\lambda(x, u)$  e emitirá a saída  $\eta(x, u)$  (BARRETO, 1995).

Tem-se então, um sistema computacional chamado de **Hipertexto** ( $H_p$ ), definido como um autômato, da seguinte forma (ALMEIDA, 1999):

$H_p = \{U, Y, X, x_0, p, \delta\}$ , onde:

$U$  é o alfabeto de entrada;

$Y$  é o alfabeto de saída;

$X$  é o espaço de estado finito;

$x_0 \in X_0 \subset X$  é o estado inicial;

$p: U \times X \rightarrow X$  é a função de transição de estados;

$\delta: U \times X \rightarrow Y$  é a função de saída do próximo estado.

O alfabeto de entrada é o conjunto de valores que o usuário pode introduzir no sistema, através de qualquer dispositivo (teclado, *mouse* etc.).

O conceito de estado caracteriza o aspecto dinâmico de hipertexto. Se cada nó de informação é associado, por exemplo, com uma janela na tela do computador, então um conjunto de janelas na tela caracteriza um estado  $x$  do hipertexto  $H_p$ .

O estado inicial  $x_0 \in X_0 \subset X$  do hipertexto  $H_p$  é qualquer estado no qual o usuário pode começar a navegar no documento. Pode existir um ou mais estados iniciais dos quais, por exemplo, um pode iniciar a localização de uma base de dados.  $X_0$  é o conjunto de todos os estados iniciais possíveis e um subconjunto de todos os estados possíveis.

A função de transição pode ser referida como uma ligação (*link*) que conecta o hipertexto como um todo. A função de transição  $\rho$  do hipertexto  $H_p$  é a função que conduz o sistema hipertexto de um estado ao outro, dada uma sequência de dados de entrada. O usuário poderá notar a transição de um estado a outro se o conjunto de nós apresentados a ele em um novo estado tem valores de saída diferentes do estado anterior.

O alfabeto de saída  $Y$  de um hipertexto  $H_p$  inclui todas as formas de representação da informação que são vistas pelo usuário na tela. O tamanho e a forma de cada janela na tela, sua posições relativas, gráficos, sons, vídeo etc, caracterizam um alfabeto de saída.

A função de transição  $\delta$  de um hipertexto  $H_p$  é uma função que, dada uma sequência de entrada e um estado, oferece a informação e sua representação na tela ou em qualquer outro dispositivo. As possibilidades de apresentação dependerão da riqueza do alfabeto de saída. A função  $\delta$  possui valores no alfabeto de saída, um conjunto de meios de apresentação que depende do estado dado e do alfabeto de entrada.

Se cada nó de informação é associado, por exemplo, com uma janela na tela do computador (uma forma possível de entrada), então um conjunto de janelas na tela caracteriza um estado  $x$  do hipertexto.

Exemplos de hipertextos como autômatos podem ser vistos em Almeida (1999).

**Outros trabalhos importantes** complementam a visão colocada sobre o uso do conceito de autômato na formalização de hipertexto. Stotts et al. (1998), prova diversas propriedades de hipertexto através de uma máquina abstrata denominada “link-automaton”. Tompa (1989) formaliza hipertextos como grafos.

### 4.3 Conclusão do capítulo

Analisando o que foi apresentado, verifica-se que a utilização de uma arquitetura baseada em **autômatos finitos com saída** tem grandes potencialidades e apresenta muitas vantagens que podem ser aproveitadas em sistemas de hipermídia adaptativa (SHA). A independência do material instrucional (modelo do domínio) em relação ao esquema de navegação, abre espaço para diversas possibilidades em SHA, como por exemplo:

- É possível desenvolver funções de saída estendidas que, controladas por algum mecanismo de adaptação, sejam capazes de gerar páginas adaptadas ou ainda efetuar operações de manuseio de links do tipo anotação, habilitação, desabilitação etc.
- Outras possibilidades envolvem a criação de agentes para registrar as atividades dos alunos e selecionar o autômato a ser disponibilizado, de acordo com regras específicas.

No contexto deste trabalho serão analisadas com maior profundidade as possibilidades de mapeamento do modelo do domínio de SHA para autômatos com saída, bem como a utilização destes no processo de adaptação.

## **5 AGENTES**

### **5.1 Introdução**

O conceito de agente tem merecido, nos últimos anos, grande atenção de pesquisadores da área de computação. A evolução rápida dos computadores e a sua crescente utilização nos diversos setores da atividade humana refletem-se em mudanças significativas no mundo dos negócios, nos meios de comunicação, enfim, na sociedade como um todo. Os computadores são colocados para resolver problemas cada vez mais complexos, requerendo soluções de software sempre mais sofisticadas.

Neste contexto entram os softwares agentes como entidades autônomas que podem colaborar entre si para resolver problemas complexos. A concepção pode ser vista como a aplicação da técnica “dividir para conquistar”, muito utilizada em computação na solução de problemas do mundo real.

John MacCarthy e Oliver Selfridge (KAY apud BRADSHAW, 1997) são considerados os precursores da idéia de agentes. O que eles pretendiam, ainda na década de 1950, era construir um sistema que apresentasse algum grau de autonomia e inteligência executando ações para alcançar uma meta, pedindo auxílio ao humano quando necessário. Um agente funcionaria como um robô residindo e atuando num mundo computacional.

Os primeiros agentes construídos resolviam problemas através de heurísticas e métodos baseados em conhecimento, ainda presentes em aplicações de Inteligência Artificial (IA) (COSTA, 1999).

As pesquisas na área de agentes demarcam duas fases distintas (NWANA, 1996). A primeira teve início em 1977, baseada principalmente em pesquisas da área de Inteligência Artificial Distribuída (IAD). Surge o conceito de agente deliberativo, que possui modelos

internos simbólicos e no qual estão presentes também aspectos como a interação e comunicação entre agentes, decomposição de tarefas, cooperação, entre outros.

A segunda fase teve início em 1990, enfatizando as características de autonomia e inteligência com base no processo de estímulo–resposta e favorecendo maior capacidade de ação efetiva local ou remota.

Avanços recentes no campo de ambientes de aprendizagem inteligentes, tem proposto a utilização de arquiteturas baseadas em sociedades de agentes. Sistemas multiagentes têm mostrado grande potencial como suporte ao processo de ensino e aprendizagem, pois podem trabalhar de forma cooperativa.

## **5.2 A inteligência artificial distribuída (IAD)**

O foco da IAD é a construção de agentes inteligentes, que podem ser vistos como entidades autônomas dentro de um sistema distribuído inteligente. Nessa forma, os agentes podem se comunicar e cooperar entre si através de algum mecanismo de natureza inteligente, na resolução de um determinado problema. A IAD trabalha na construção da inteligência coletiva, o que a diferencia da IA tradicional a qual trabalha na construção da inteligência individual. Nesse enfoque, a IAD procura resolver problemas complexos e de maior porte nos quais a solução depende, fortemente, de aspectos cooperativos entre as entidades envolvidas na busca da solução global, podendo ocorrer inclusive o gerenciamento de dados distribuídos fisicamente.

A Inteligência Artificial Distribuída pode ser dividida nas seguintes áreas (SICHTMAN et al., 1995):

- Sistemas Multiagentes (MAS)
- Resolução Distribuída de Problemas (DSP)

A DSP tem como objetivo a solução de um problema particular. Projeta-se a arquitetura para resolver especificamente esse tipo de problema, sendo os agentes programados para realizar tarefas específicas e interagir mutuamente na busca da solução global. O número de agentes é fixo e cada um possui uma visão específica e parcial do problema.

Os sistemas multiagentes (MAS), são projetados para resolver problemas antes desconhecidos e não simplesmente um problema específico como acontece em DSP. Assim, nessa abordagem ocorre primeiramente a concepção dos agentes para que depois possa ser estudado o ambiente em que poderão atuar. Um aspecto importante em MAS é que os

agentes componentes devem ser capazes de identificar modificações ocorridas no ambiente, alterando a sua representação interna.

Percebe-se claramente que a arquitetura MAS apresenta maior flexibilidade pela maior abrangência. No entanto há com certeza uma sobrecarga de processamento decorrente da maior complexidade das tarefas no processo de interação e cooperação entre agentes.

### 5.3 Definição de agentes

Na literatura não se encontra uma definição universal sobre “O que é um agente?”. Os termos **agência** e **autonomia** são utilizados com crescente frequência para especificar diferentes noções, com diferentes conotações. Os pesquisadores colocam a sua visão sobre o tema de acordo com suas experiências na modelagem e implementação de agentes em aplicações diversas.

Assim, no mundo computacional visto sob a ótica da engenharia de software, os agentes interagem de forma mais ou menos inteligente com outros softwares, executando tarefas diversas com certo grau de autonomia e inteligência. Como exemplos, pode-se citar agentes de rastreamento, agentes de informação, agentes de gerenciamento de banco de dados, agentes de busca na Internet.

No contexto cognitivo a visão é voltada para tentar simular, modelar e até mesmo explicar funções mentais humanas. De um modo geral procura-se criar para esses agentes um ambiente com elementos pré-selecionados do mundo real, fazendo com que eles ajam de forma autônoma ou suportando a intervenção humana.

Em ambos os casos citados os agentes devem conhecer o domínio de suas atividades e escolher e/ou planejar ações.

Apresenta-se a seguir a visão de pesquisadores sobre o conceito de agente.

Segundo Maes (1995), agentes são sistemas computacionais residentes em ambientes dinâmicos complexos nos quais captam informações e atuam de forma autônoma, realizando um conjunto de metas e tarefas para as quais foram projetados.

Shoham (1997, p. 75) define agente como: “uma entidade que funciona continuamente e de forma autônoma em um ambiente particular, freqüentemente habitado por outros agentes e processos“. O requisito de continuidade e autonomia é necessário para garantir a atuação do agente de forma mais independente possível do usuário. É desejável também que um agente, funcionando de forma contínua em um ambiente por um longo período, seja capaz de aprender com suas experiências. Além disso, espera-se que tal agente seja capaz de se

comunicar com outros agentes e processos de forma a cooperar com eles e até mesmo mover-se de um lugar para outro.

Segundo Norvig e Russel (1995), um agente é uma entidade que pode perceber o seu ambiente através de sensores ou sistemas de percepção, e atuar sobre ele através de mecanismos de atuação (atuadores). Para isso dispõe de uma representação parcial desse ambiente, podendo se comunicar com outros agentes no caso de pertencer a uma sociedade de agentes.

Wooldridge e Jennings (1995) conceituam agentes sob dois enfoques: noção de agência fraca e noção de agência forte.

No primeiro, um agente é visto como um sistema computacional baseado em hardware e/ou software que possui propriedades tais como: autonomia, sociabilidade, reatividade e proatividade. No segundo, um agente é uma entidade que possui adicionalmente qualidades mentais ou emocionais explícitas: conhecimento, crenças, intenção e obrigações.

Caglayan e Harrison (1997), visualizam um agente como um modelo no qual é evidenciada a orientação por tarefas e o suporte à inteligência distribuída. A figura 6 apresenta um modelo de agentes.

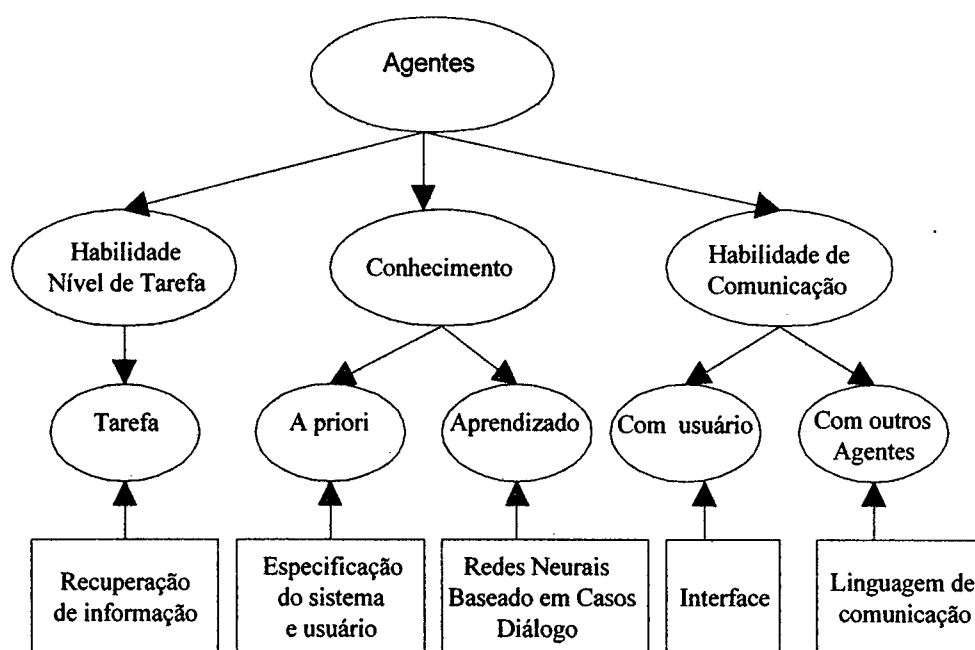


Figura 6 – Modelo de Agentes (CAGLAYAN; HARRISON, 1997).

De acordo com a figura 6, o modelo possui três níveis, que são:

1. **as habilidades na execução de tarefas (nível de tarefas)** - o nível de tarefas especifica a funcionalidade de um agente, envolvendo a sua capacidade de perceber o ambiente no qual está localizado através de sensores e atuar sobre esse ambiente para executar as tarefas e alcançar metas;
2. **conhecimento** – normalmente um agente possui um conjunto de regras para completar suas tarefas, que define o seu conhecimento sobre o ambiente. A aquisição do conhecimento pode ser feita de várias formas:
  - o conhecimento é organizado a priori, de maneira formal, através esquemas típicos da área de IA, em tempo de projeto e desenvolvimento;
  - o conhecimento é especificado diretamente pelo usuário através de regras programadas por ele para dar suporte às tarefas a serem cumpridas;
  - o conhecimento é derivado de outras fontes, como por exemplo, através de linguagens comuns de agentes que possibilitam a aquisição do conhecimento entre comunidades de agentes;
  - aprendizado pelo sistema: agentes que derivam conhecimento do usuário e do ambiente.
3. **habilidades de comunicação** – dois aspectos devem ser considerados. O primeiro se refere à interface com o usuário, que deve ser projetada aproveitando-se as funcionalidades previstas para o agente. O segundo trata da comunicação com outros agentes, realizada através de uma linguagem de comunicação própria.

## 5.4 Propriedades dos agentes

Na literatura percebe-se que os pesquisadores procuram identificar um agente através de propriedades básicas. Observa-se que um agente não precisa apresentar todas as propriedades, pois cabe ao projetista incluí-las numa aplicação conforme as funcionalidades requeridas. As propriedades mais relevantes são listadas a seguir:



- **autonomia:** capacidade do agente em agir sozinho, não necessitando da orientação de usuários ou outros agentes, ou seja, o agente possui o controle sobre suas ações no cumprimento de suas metas (MAES, 1994).
- **comunicabilidade:** é a capacidade do agente de se comunicar com outros agentes ou pessoas no ambiente no qual está inserido (FRANKLIN; GRAESSER, 1996);
- **reatividade:** é a propriedade que possibilita ao agente perceber o seu ambiente e atuar sobre ele. É uma propriedade essencial de um agente inteligente. O agente deve possuir sensores e um modelo interno do ambiente que permita reagir a mudanças no mesmo (NWANA, 1996).
- **pró-atividade:** através desta propriedade o agente não age simplesmente em resposta ao ambiente, mas toma iniciativas sobre circunstâncias específicas. Para que esta habilidade seja alcançada é necessário que o agente possua metas bem definidas, em outras palavras, que ele seja orientado a metas (NWANA, 1996);
- **cooperação:** Uma extensão natural da propriedade de comunicação é a cooperação. É a habilidade através da qual vários agentes instituem um diálogo entre si e possivelmente com humanos, através de alguma linguagem de comunicação, com intuito de atingir determinada meta (WOOLDRIGE; JENNINGS, 1995);
- **continuidade:** agentes são processos que estão em processamento contínuo, na forma ativa ou passiva (“sleeping”);
- **mobilidade:** é a capacidade do agente de navegar em uma rede de computadores, podendo ir de um computador para outro. Esta propriedade envolve questões não muito bem resolvidas como segurança, privacidade de dados, gerenciamento entre outras.
- **aprendizagem:** um agente inteligente deve apresentar pelo menos capacidade de alterar seu comportamento baseado em experiências prévias (FRANKLIN; GRAESSER, 1996).
- **inteligência:** é demonstrada pelo agente a partir de sua habilidade de negociação efetiva com ambigüidades e com a capacidade de inferência, usando alguma forma de abstração simbólica (SHOHAM, 1993).
- **veracidade:** suposição de que um agente não comunicará informações falsas.

## 5.5 Arquiteturas de agentes

A arquitetura de um agente especifica a sua estrutura interna e funcionamento. Em outras palavras, a arquitetura de um agente especifica como podem ser estruturados os módulos componentes do agente e como estes módulos devem ser construídos para interagir. O conjunto total de módulos e suas interações devem prover uma resposta para a questão de como os sensores de dados e o estado interno corrente determinam as ações e o seu estado futuro (MAES, 1991).

O esquema arquitetônico pode ser pensado, segundo Brenner (1998), como composto de três módulos: módulo responsável pela entrada (percepção), módulo de processamento inteligente e módulo de saída (ações).

O módulo de entrada obtém informações do ambiente percebido e provê subsídios para o alcance das metas. O módulo central é responsável pelo processamento das informações obtidas na entrada e pode possuir refinamentos de acordo com o nível de percepções possíveis. O módulo de saída trata das ações a serem executadas pelo agente para o alcance e manutenção de suas metas.

Para Wooldridge e Jennings (1995), as arquiteturas de agentes podem ser divididas em três áreas:

- **arquitetura cognitiva ou deliberativa:** nesta arquitetura os agentes possuem modelos simbólicos explícitos de seus ambientes nos quais as ações são tomadas via raciocínio lógico, ou seja, os agentes são vistos como parte de um sistema baseado em conhecimento. As dificuldades encontradas nesse tipo de arquitetura são duas: em primeiro lugar, a tradução do mundo real em uma descrição simbólica, de forma exata e adequada, é difícil e pode tomar muito tempo; em segundo lugar, há o problema de como representar simbolicamente as informações sobre entidades e processos de forma que os agentes possam raciocinar com essas informações, em tempo hábil.
- **arquitetura reativa:** Uma arquitetura reativa não inclui nenhum tipo de modelo do mundo simbólico pré-estabelecido e não utiliza raciocínio simbólico complexo. A decisão a tomar é implementada através de algum mapeamento do tipo situação-ação;
- **arquitetura híbrida:** nessa arquitetura há uma combinação das duas abordagens citadas anteriormente. Essa arquitetura, também chamada de “arquitetura em

camadas”, estrutura um agente com dois subsistemas: um deliberativo contendo o modelo simbólico e outro reativo, capaz de reagir a eventos que ocorram no ambiente.

Weiss (1999) apresenta uma classificação que se aproxima da proposta por Wooldridge e Jennings, incluindo a arquitetura BDI (agentes com crenças, desejos e intenções) que pode ser vista em Giraffa (1999).

## **5.6. A Sociedade de agentes**

Agentes trabalhando em sociedade, principalmente na modalidade dos chamados sistemas multiagentes, reforçam a funcionalidade da tecnologia de agentes, constituindo-se em ferramenta poderosa para a busca de soluções para problemas de natureza distribuída.

Numa sociedade, um agente ao realizar uma determinada tarefa geralmente necessita da ajuda de outros agentes, ou então atua no auxílio de outros para satisfazer o objetivo geral. Dessa forma, os agentes precisam saber como encontrar alguém com quem cooperar visando a busca desse objetivo. Algumas abordagens de arquiteturas de sociedades de agentes são apresentadas a seguir, conforme Faraco (1998).

### **5.6.1 Arquitetura quadro negro (blackboard)**

Nesta arquitetura, todas as interações entre os agentes acontecem através de uma estrutura de dados normalmente persistente denominada quadro negro. Os agentes lêem e escrevem suas mensagens no quadro negro quando necessário. Pode-se afirmar que um quadro negro é uma memória de compartilhamento global, na qual são depositadas informações de interesse da sociedade de agentes. Em sistemas multiagentes o quadro-negro pode ser utilizado como um repositório de perguntas e respostas. Os agentes que necessitam alguma informação escrevem seu pedido no quadro, que quando inspecionado pelos outros agentes, responderão para uma posterior recuperação dessa resposta pelo agente que questionou.

### **5.6.2 Arquitetura de troca de mensagens**

Nesta arquitetura os agentes comunicam-se diretamente uns com os outros enviando mensagens assíncronas. Não existe o papel do quadro negro como intermediário do processo de interação, mas pode haver a presença de um agente facilitador de comunicação. Esse tipo de arquitetura exige que os agentes saibam os nomes e endereços uns dos outros para que as mensagens sejam devidamente encaminhadas. Esse método é mais eficiente para se obter as mensagens em tempo hábil, mas por outro lado, o projeto deve prever estudos sobre a relação entre o número de agentes ativos e o número de mensagens, para evitar congestionamento no tráfego de mensagens. Para que as trocas de mensagens ocorram de maneira adequada entre os agentes é necessário estabelecer um protocolo de conversação. O protocolo é quem dita as regras e impõe o formalismo necessário para que as mensagens sejam encaminhadas e compreendidas pelos agentes.

### **5.6.3 Arquitetura federativa**

Esta arquitetura procura resolver o problema da comunicação interna entre agentes, principalmente quando fica crítica a utilização da arquitetura anterior (message passing). Na arquitetura federativa os agentes da sociedade são divididos em grupos ou federações segundo um critério de agrupamento escolhido. Junto a cada grupo de agentes encontram-se os agentes facilitadores, responsáveis por receber a mensagem que chega em cada grupo e encaminhá-la para o agente destinatário presente naquele grupo. A vantagem dessa arquitetura é que o agente facilitador tem a propriedade de identificar se a mensagem que chega é destinada a algum agente de seu grupo e, se for o caso, fazer a devida entrega.

A arquitetura federativa propõe a diminuição do fluxo de mensagens desnecessárias entre os agentes que formam a sociedade, pois os facilitadores têm a capacidade de remetê-las ao respectivo destinatário sem a necessidade de enviá-las a todos os agentes.

## **5.7 Agentes e o processo pedagógico**

Segundo Giraffa (1999) a utilização do conceito de sociedade de agentes em sistemas voltados para a educação assume grande importância atualmente. Recentemente a literatura

tem apresentado a noção de **agentes pedagógicos** tendo em vista que muitos sistemas com fins educacionais tem adotado o paradigma dos agentes. Agentes pedagógicos podem ser do tipo *goal driven*, dirigidos por objetivos, como por exemplo, tutores, mentores e assistentes, e do tipo *utility-driven*, guiados pela sua utilidade no ambiente, como por exemplo, agentes da WEB.

Os agentes *utility driven*, são usados como auxiliares de professores e alunos na execução de tarefas ligadas às atividades pedagógicas. Ajudam os estudantes a encontrar material didático (arquivos, diretórios, programas etc.), agendam encontros entre grupos, lembram datas de entrega de trabalhos, entre outras atividades.

Os agentes *goal driven* na WEB possuem mobilidade e operam em diferentes contextos como por exemplo, contextos textuais, hipermídia e realidade virtual. Neste caso a interação ocorre baseada no processo cooperativo ou competitivo no qual os agentes artificiais e agentes humanos se comunicam e realizam atividades.

A introdução de agentes em Sistemas Tutoriais Inteligentes (STI) permite ganhos de qualidade sob o ponto de vista pedagógico (GIRAFFA, 1999). Fatores como pro-atividade, habilidade social e flexibilidade são mais facilmente incorporadas através de agentes, atendendo de maneira mais efetiva as necessidades do STI.

Segundo Vassileva (apud GIRAFFA, 1999), o relacionamento entre agentes pode ser visto como uma nova fonte de recursos para atingir os objetivos de um STI. Os agentes pedagógicos podem atuar como tutores virtuais, estudantes virtuais ou companheiros virtuais de aprendizagem. As propriedades fundamentais de agentes pedagógicos são: autonomia, habilidade social, pro-atividade e persistência. Além destas propriedades os agentes pedagógicos podem ser reativos, de performance contínua, serem capazes de aprender e na maioria dos casos, serem representados por um personagem.

Como exemplo de Sistemas Tutores Inteligentes que utilizam uma arquitetura multi-agentes, pode-se citar o Sistema Mathema e o Sistema Ame-a, cujas descrições podem ser vistas em Giraffa (1999).

## **5.8 Comunicação entre agentes.**

A habilidade de comunicação é um atributo importante que os agentes devem possuir. Num sistema moderno de computação, frequentemente temos vários computadores interagindo de forma distribuída, nesse contexto torna-se clara a necessidade de uma

linguagem de comunicação comum. Atualmente não existe uma linguagem padrão aceita mundialmente para a representação de informações trocadas por agentes.

Segundo Costa (1999), o projeto de uma linguagem de comunicação entre agentes pode seguir uma abordagem procedural ou declarativa. Na abordagem procedural a comunicação acontece através de diretivas. Linguagens baseadas em *scripts*, tais como TCL, Java e Telescript são exemplos da abordagem procedural. Na abordagem declarativa o enfoque é a comunicação através da troca de estruturas declarativas, tais como definições, asserções e outras, tendo como base a lógica dos predicados de primeira ordem.

O grupo DARPA KSE (*Knowledge Sharing Effort*) usa uma abordagem declarativa. Este grupo definiu uma linguagem de comunicação de agentes denominada ACL, que pode ser dividida em três partes (COSTA, 1999):

- **vocabulário** – é um grande dicionário de palavras associadas às áreas de aplicação comum. Esse vocabulário compartilhado é chamado de Ontologia;
- **linguagem Interna** – KIF (*Knowledge Interchange Format*) possui a capacidade de representar dados, restrições, regras, expressões e outros, utilizando uma versão do cálculo de predicados de primeira ordem;
- **linguagem Externa** – KQML (*Knowledge Query and Manipulation Language*) permite uma comunicação eficiente podendo inclusive encapsular estruturas KIF.

Uma mensagem em ACL é uma expressão KQML na qual os argumentos são termos ou sentenças em KIF, formadas por palavras encontradas no vocabulário da ACL. O dicionário é aberto, permitindo o acréscimo de palavras nas áreas já existentes ou então em novas áreas que possam surgir. Têm-se dessa maneira uma linguagem padrão de comunicação entre agentes.

Como fica a questão da interoperabilidade, ou seja, a convivência entre os programas existentes e os agentes que se pretende implantar numa aplicação?

Para resolver essa questão, pode ser usada a seguinte abordagem, segundo Genesereth e Ketchpel (1994):

- **utilização de Transdutor** – a aplicação envia a mensagem, o transdutor transforma em ACL e passa ao agente. Quando da resposta, o transdutor converte na linguagem da aplicação. A vantagem desta abordagem é que ela pode funcionar em qualquer aplicativo; como desvantagem poderá ocorrer um tempo de resposta mais lento;

- **utilização de Empacotador** – nessa abordagem é inserido código na aplicação para permitir a comunicação em ACL. Esse empacotador pode examinar estruturas de dados no programa e modificá-las. Torna a comunicação mais eficiente por não ter de traduzir e diminuir a comunicação serial, porém necessita que o código original esteja disponível;
- **reescrita** – é a opção mais drástica que envolve reescrever toda a aplicação na linguagem ACL.

## 5.9 Conclusão do capítulo

A utilização do conceito de agentes na busca de soluções para sistemas distribuídos, de um modo geral, é bastante promissora e está evoluindo rapidamente.

Como foi visto, os agentes são entidades que podem se comunicar e cooperar entre si através de algum mecanismo de natureza inteligente, na resolução de um determinado problema. Em ambientes de aprendizagem, fatores como pro-atividade e habilidade social são muito importantes e são mais facilmente incorporados através dos agentes. Além destas propriedades os agentes pedagógicos podem ser reativos, de performance contínua, serem capazes de aprender e na maioria dos casos, serem representados por um personagem.

Propostas recentes de utilização de arquiteturas de sistemas multiagentes em ambientes de aprendizagem em geral, apresentam muitas vantagens em relação às arquiteturas tradicionais. Pode-se citar ganhos em flexibilidade no tratamento dos elementos que compõem o sistema, construção por módulos quase independentes (modularidade), possibilidade de dividir o problema em sub-problemas e confiança no sentido de que a resolução de um problema pode continuar ainda que haja falha em um dos agentes.

Em termos operacionais os agentes oferecem as seguintes vantagens: eficiência (tendem a consumir menos recursos de rede), menor tráfego em rede (utilizando agentes móveis), interação autônoma assíncrona (as tarefas podem ser codificadas nos próprios agentes e então serem despachadas), robustez e tolerância à falhas, adaptação dinâmica ao ambiente, capacidade para trabalhar com grande volume de dados (com deslocamento do agente para servidores de banco de dados, evitando grandes volumes de transferência de dados via rede) e suporte a ambientes heterogêneos.

## 6 INTELIGÊNCIA ARTIFICIAL APLICADA NA EDUCAÇÃO (IAED)

### 6.1 Introdução

A aplicação de técnicas de inteligência artificial na solução de problemas educacionais constitui a área de pesquisa conhecida como IAED. Sistemas desenvolvidos nesta área possuem algum grau de tomada de decisão em relação às suas interações com os usuários, que na maioria das vezes são estudantes. O processo de decisão é realizado on-line requerendo agilidade do sistema no acesso à base de conhecimentos e aos processos de raciocínio. As fontes da IAED têm relação com várias áreas do conhecimento humano, tais como Ciência da Computação, Psicologia e Educação (ROSATELLI, 2000).

Para Balasubramaniam (1994) uma base de conhecimento e uma máquina de inferência construída no topo da base de dados hipermídia, pode adicionar inteligência aos nós e *links*. Jonassen (1988), propõe duas soluções: software de integração de rede semântica com hipermídia, como uma estratégia de aprendizagem cognitiva e um sistema baseado em conhecimento como *front-end*, sendo utilizado para assistir os alunos no acesso às informações.

Segundo Santos et al. (1996), os sistemas com navegação assistida, com maior ou menor grau de refinamento, precisam acessar informações do modelo do usuário e da base de conhecimentos. Nesse contexto, pelo menos duas propostas de navegação assistida agregando componentes baseados em conhecimento estão relatadas em: RRI – *Research Reference Interface* (ZHONGMIN et al., 1991) e SINS – *Semistruture Navigation System* (FUJIHARA et al., 1992). Estas ferramentas são utilizadas em sistemas especialistas, sistemas de ajuda inteligentes e sistemas de hipertexto.



A utilização de técnicas de IA na construção de *softwares* educacionais pode ser vista como fator de incremento de qualidade. Giraffa (1999) destaca os seguintes aspectos relacionados com a qualidade de sistemas educacionais: os sistemas necessitam incorporar componentes de IA cujo desenvolvimento está longe do trivial; a representação do conhecimento (modelagem do domínio), principalmente no que se refere à habilidades trabalhadas na forma de procedimentos, não é ainda uma solução satisfatória; os sistemas necessitam de capacidades que permitam construir uma estrutura com estratégia explícita de navegação, selecionada a partir do aluno, intensificando dessa forma aspectos pedagógicos no ambiente; é importante tornar o conhecimento explícito e modificável para permitir ao estudante maior interação com o sistema.

Constata-se que apesar da grande evolução que a IAED teve nos últimos anos, há muito espaço de pesquisa a ser percorrido para que sistemas voltados para a educação possam atender de forma efetiva e eficaz os requisitos de qualidade.

## 6.2 Histórico

Segundo Beck et al. (1998), os primeiros sistemas, desenvolvidos nos últimos 20 anos, foram conhecidos como Treinamento Baseado em Computador (CBT – *Computer Based Training*) e Instrução Auxiliada por Computador (CAI – *Computer Assisted Instructional*), os quais são referenciados pela grande maioria da literatura simplesmente como CAI. Nestes sistemas, o material didático é organizado conforme o paradigma da instrução programada, sendo dividido em segmentos de informação e problemas pertinentes, a serem seguidos passo a passo. Em cada passo o sistema avalia as soluções apresentadas pelo usuário aluno e dependendo do grau de acerto é indicado o próximo passo. Os sistemas CAI evoluíram nos últimos anos sendo que muitos deles permitem a utilização de recursos de hipermídia e podem ser utilizados via Web.

Os sistemas CAI não consideram as necessidades individuais do usuário aluno bem como não oferecem ao mesmo o tipo de atenção que poderia ser oferecido por um tutor humano. Para melhor atender o estudante, um sistema CAI teria que possuir mecanismos que permitissem o raciocínio sobre o domínio e sobre o modelo do estudante.

Pesquisas realizadas a partir da década de 70, visando a aplicação de técnicas de IA em sistemas CAI, originaram os sistemas denominados ICAI (*Intelligent Computer Assisted Instructional*), que alguns autores classificam como Sistemas Tutoriais Inteligentes (STI).

Nos STI pretende-se que o computador mantenha uma conduta inteligente que permita controlar o processo de aquisição do conhecimento por parte do usuário aluno favorecendo um diálogo aberto (GIRAFFA, 1999).

Segundo Rosatelli (2000), o paradigma dos STI dominou a área de IAED por bastante tempo. Estes sistemas partiam da idéia de implementação dos métodos de ensino e aprendizagem baseados no currículo escolar tradicional, cujos objetivos podiam ser mensurados pelos testes padrão existentes. Os STI inicialmente foram projetados com a função maior de ajudar o estudante na resolução de problemas num determinado domínio do conhecimento, possuindo pouca referência ao material didático *on-line*. A partir dos anos 80, novos paradigmas de instrução têm surgido, refletindo mudanças de enfoque na questão da educação, originando soluções conhecidas como ambientes de aprendizagem interativos e micromundos. Estas soluções procuram implementar o processo de ensino e aprendizado baseado na investigação, e colaboração, colocando o aluno no centro do processo. Como consequência, os STI, na sua concepção original, começam a ser considerados limitados.

Por outro lado fica cada vez mais claro que o hipertexto ou hipermídia prove melhores opções para a organização do material de ensino *on-line*, inclusive de forma mais abrangente com aplicações na *Web*. A junção dos recursos de STI com o material de ensino organizado como hipermídia foi o ponto de partida para os sistemas de hipermídia adaptativa aplicados na educação (BRUSILOWSKY, 1999), na última década.

Pesquisas mais recentes na área de IAED indicam a utilização de arquiteturas baseadas em sistemas multiagentes para a implementação de ambientes de aprendizagem, como forma de se obter ganhos de qualidade nos STI bem como nos SHA já referenciados anteriormente.

### **6.3 Questões relevantes na área de IAED**

Self (apud ROSATELLI, 2000), coloca e discute as seguintes questões que são relevantes na discussão de problemas educacionais, no contexto da IAED:

#### **6.3.1 Natureza do conhecimento**

A natureza do conhecimento pode ser vista sob a ótica das seguintes filosofias:

- **objetivismo:** considera que o mundo pode ser estruturado em termos de entidades, propriedades e relacionamentos e que é possível raciocinar através da manipulação de símbolos abstratos que representam a realidade;
- **construtivismo:** é o oposto do objetivismo e defende que o conhecimento é imposto no mundo por nós ao invés de existir no mundo, independente de nós. Nessa perspectiva um sistema de IAED não conteria conhecimento correto *a priori*, mas teria que descobri-lo juntamente com o usuário aluno;
- **situacionismo:** afirma que o conhecimento não existe na memória, mas surge através da interação com o ambiente. Os situacionistas argumentam que as representações do conhecimento são criadas no curso de atividade, mas não são por si só conhecimento, o qual estaria então na capacidade de interação;
- **conexionismo:** é outra visão do conhecimento que contrasta com o processamento de símbolos relativo ao objetivismo. O conexionismo afirma que o conhecimento está implicitamente representado através de pesos e ligações entre um grande número de nós modelados em redes neurais. Os métodos do conexionismo foram usados desenvolver componentes de sistemas IAED, mas nenhum sistema deste tipo segue inteiramente uma filosofia de conexionismo, isso porque a representação implícita do conhecimento não suporta, de maneira fácil, interações de aprendizagem.

### 6.3.2 Natureza do aprendizado

Na área de IAED há muitas dificuldades de pesquisa porque não existe um mapeamento inequívoco entre as filosofias de conhecimento e filosofias de aprendizagem.

Em geral, o *design* de um sistema reflete uma visão um tanto eclética da natureza do aprendizado. O conhecimento nos sistemas IAED pode ser obtido através de diversas maneiras. As que atualmente mais são utilizadas estão enumeradas abaixo:

1. **aprendizagem dirigida por falhas:** baseada na hipótese que a ocorrência de falhas proporciona as oportunidades para o aprendizado;
2. **aprendizagem baseada em casos:** engloba a idéia de que os estudantes aprendem a partir de histórias (casos) apresentados no exato momento em que necessitam da

informação trazida pelo caso. Essa forma de aprendizado tem origem nas teorias de IA que tratam de raciocínio baseado em casos.

3. **aprendizagem por experimentação/simulação:** resolução de problemas em que os estudantes realizam experimentos e são guiados pela suas interpretações;
4. **aprendizagem através do diálogo:** defende que existe uma necessidade de engajar o estudante num diálogo, de forma a captar-se as concepções errôneas. Tais concepções errôneas podem ser derivadas, por exemplo, da interação dos estudantes com um sistema objetivando um efeito de curto prazo, sem, no entanto, enfocar as suas crenças e concepções equivocadas, que por sua vez, permanecerão no longo prazo;
5. **aprendizagem como uma atividade social:** propõe que a ênfase em projetar um sistema deve ser no contexto social e físico no qual ele será utilizado;
6. **aprendizado por colaboração:** propõem que o aluno possa aprender através da colaboração com outro aluno;
7. **aprendizado por necessidade:** situa o aprendizado num contexto de trabalho ao invés de colocá-lo numa fase separada, fazendo com que o aprendizado seja relevante à tarefa em execução.

### 6.3.3 Estilos de ensino

Vários estilos de ensino são utilizados em sistemas IAED. A maioria deles é uma investigação de um dos estilos de ensino aplicado a um tópico específico. Em tais sistemas o equilíbrio entre “guiar”, “instruir” ou “deixar” o estudante e, conseqüentemente, toda a questão do equilíbrio do controle entre o estudante e o sistema, é um debate continuo na área de IAED. As dificuldades de se utilizar de forma adequada e muitas vezes integrada os diferentes estilos de ensino, tem relação com: primeiro, a concepção ingênua de que a especialidade do professor pode ser embutida nos sistemas IAED; segundo, os designers de sistemas IAED não são especialistas no domínio ou não tem acesso a tal especialidade; terceiro, o componente pedagógico tem sido considerado de menos importância do que as representações do conhecimento sobre o domínio.

### **6.3.4 Novas tecnologias na educação**

As técnicas de IA podem ser adotadas para superar as limitações de aplicação das novas tecnologias na educação. Assinala-se a seguir as principais tendências:

- o uso de interface multimídia que requer modelos da mídia, do usuário, da tarefa e do discurso;
- a efetividade da realidade virtual como ambiente de aprendizagem;
- introdução de agentes inteligentes no ambiente;
- a demanda por ambientes colaborativos apoiados por computador para treinamento no ambiente de trabalho.

### **6.3.5 Medidas de eficácia de um sistema**

A pesquisa de IA aplicada na educação deve ser avaliada para que se possa garantir uma evolução na área de IAED. Dois tipos de avaliação podem ser propostos: a avaliação de sistemas completos, que tenta mostrar os benefícios do sistema como um todo; avaliação de componentes ou protótipos do sistema, que objetiva investigar as propriedades de partes do sistema tal que esta parte possa ser melhorada.

## **6.4 Sistemas Tutores Inteligentes**

### **6.4.1 Introdução**

Segundo Rosatelli (2000), a aplicação mais típica da IAED são os Sistemas Tutores Inteligentes (STI), que constituem uma tentativa de implementar, num sistema computacional, os métodos tradicionais de ensino e aprendizado (entre tutor e aluno). O tutoramento um-a-um permite que o aprendizado seja altamente individualizado e, conseqüentemente, permite um melhor resultado. Um STI é baseado na hipótese de que o conhecimento de um estudante pode ser modelado, rastreado e corrigido.

### 6.4.2 Conceituação

Segundo Hall (apud COSTA, 1999), os STI são uma composição de diversas disciplinas como Psicologia, Ciência Cognitiva e Inteligência Artificial. O objetivo principal destes sistemas é a modelagem e representação do conhecimento especialista humano para auxiliar o estudante através de um processo interativo.

Segundo Viccari (1990), os STI são programas que modificam suas bases de conhecimento, percebem as intervenções do aluno e são dotados da capacidade de aprender e adaptar suas estratégias de ensino mediante a intervenção do aluno. No processo de ensino através de um STI o estudante aprende fazendo. O sistema não ensina, o estudante é que aprende. Nestes sistemas existe uma interação entre o tutor e o estudante.

Giraffa (1999) afirma que o objetivo fundamental dos STI é de fornecer uma instrução adaptada ao aluno, tanto no conteúdo como na forma, superando os problemas mais cruciais do software educativo na atualidade. Neste sentido, os STI deveriam apresentar um comportamento próximo ao do professor humano, meta esta que está ainda longe de ser alcançada. Entre as razões para que isto ocorra pode-se citar: limitações de hardware e software que não permitem utilizar efetivamente dispositivos para o tratamento adequado de dados relativos aos sentidos (olfato, tato e visão); não se tem uma resposta de como são armazenadas, processadas e recuperadas informações no cérebro humano, informações estas que são vitais para a construção do modelo do aluno.

### 6.4.3 Componentes

A maioria dos STI segue a uma estrutura composta por quatro componentes: módulo estudante, módulo tutor (também chamado módulo pedagógico), módulo domínio (também chamado módulo especialista) e módulo de interface (também chamado módulo comunicação ou ambiente de ensino). Beck et al. (1998), numa visão mais recente, incluiu de forma separada o módulo especialista que é normalmente colocado no módulo do domínio. A seguir descreve-se a visão colocada pelo autor citado, de acordo com a figura 7.



domínio. As principais decisões tomadas neste módulo são como adquirir o conhecimento, como o mesmo ficará armazenado (ou representado) e quais são os critérios relativos ao desempenho.

#### **6.4.7 Módulo de comunicação**

O módulo de comunicação, também denominado módulo de interface ou ambiente de ensino, é a interface entre o STI e o aluno. A questão é definir qual será a melhor maneira de apresentar o material didático ao aluno. A implementação deste módulo possui complexidade variável, podendo ser desde simples janelas de diálogo até linguagem natural, reconhecimento de voz e realidade virtual.

#### **6.4.8 Módulo especialista**

O módulo especialista é similar ao domínio do conhecimento que deve conter a informação que está sendo ensinada ao aluno. Entretanto, é mais do que apenas uma representação dos dados; é um modelo de como alguém hábil em um domínio particular, representa o conhecimento. Usando o módulo especialista, o tutor pode comparar a solução do aluno à solução do especialista, localizando os assuntos de maior dificuldade para o aluno.

### **6.5 Conclusão do capítulo**

O uso de técnicas de IA tem apresentado resultados positivos em diversos softwares voltados para a área educacional. Há, no entanto, um longo caminho a trilhar para que se possa resolver, de forma satisfatória, as questões relacionadas com os processos de ensino e aprendizagem.

Percebe-se claramente que há a necessidade de se pesquisar mais aspectos psicológicos e pedagógicos envolvidos e entender melhor os aspectos cognitivos de como os estudantes aprendem.

Por outro lado, pesquisas na área de educação apontam para novos modelos de ensino mais centrados no aluno e no seu potencial de aprender, fugindo do modelo centrado no professor.



A questão da autoria de aplicações também é complexa, principalmente quando se trata de sistemas que utilizam técnicas de IA para mapear o processo de ensino e aprendizagem. Nestes casos são necessárias ferramentas que facilitem ao máximo o trabalho do professor autor na preparação do material didático. Em ambientes de aprendizagem para a Web, baseadas no currículo seqüencial tradicional, que atendem grande massa de estudantes, é conveniente implementar mecanismos que permitam o reaproveitamento do material didático em cursos com alto grau de similaridade.

A tendência percebida em IAED é que a meta não é apenas ensinar as habilidades tradicionais de forma mais rápida, eficiente e com menor custo, mas sim trabalhar com novos métodos educacionais para aplicação em sala de aula.

## **7 A ARQUITETURA PROPOSTA**

### **7.1 Introdução**

Neste capítulo, cumprindo os objetivos da Tese, é proposta uma **solução unificada** que atenda aos requisitos funcionais dos SHA, inspirada principalmente no modelo de referência para sistemas de hipermídia adaptativa AHAM (DE BRA; HOUBEN; WU, 1999), extensão do modelo Dexter (HALASZ; SCHWARTZ, 1990) e nos trabalhos de Almeida (1999), Machado et al. (1999) sobre o uso de autômatos em sistemas de hipermídia (não adaptativos).

A solução envolve autômatos finitos com saída, sociedade de agentes e conceitos de inteligência artificial para realizar o processo de adaptação.

Será apresentada uma arquitetura geral para a unificação das tecnologias e conceitos acima referidos, bem como comentários sobre o sistema protótipo construído para a avaliação e validação de algumas das estruturas propostas.

### **7.2 Autômatos e a representação do modelo do domínio.**

Inicialmente são introduzidas as primitivas de modelagem para a abordagem, envolvendo conceitos e relacionamentos. Em seguida é discutida a representação do domínio através de autômatos e a interconexão entre o modelo do domínio e as páginas hipermídia. Conforme foi visto no capítulo quatro, a formalização de um sistema de hipertexto/hipermídia pode ser feita através do conceito de autômato, trazendo uma série de vantagens para o sistema SHA nos aspectos de controle da navegação bem como na questão da adaptabilidade.

O que se pretende aqui é mostrar as questões relacionadas com o mapeamento do modelo do domínio para autômatos com saída.

### 7.2.1. Conceitos e relacionamentos entre conceitos

O modelo do domínio é o elemento central do SHA, pois nele estará representado todo o conhecimento organizado. A estruturação do material instrucional na forma hierárquica, em tópicos e subtópicos, do mais geral para o mais específico, relacionando conceitos, exemplos, exercícios, testes etc. é universalmente utilizada na área educacional porque apresenta vantagens no entendimento por parte de alunos e professores. A estrutura básica é mostrada na figura 8 (De BRA; HOUBEN; WU, 1999).

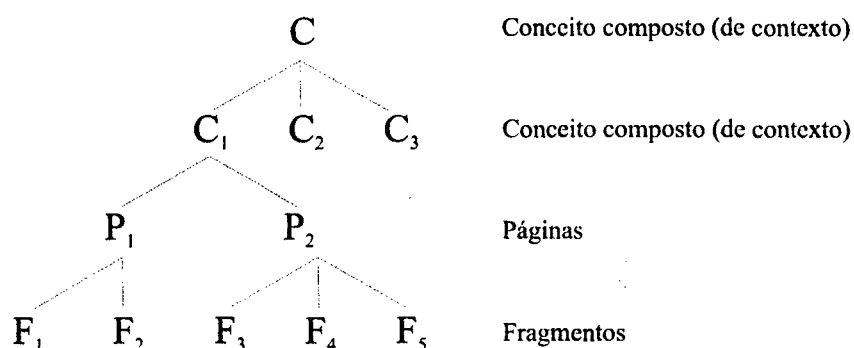


Figura 8 - hierarquia de conceitos – modelo de referência AHAM (De BRA; HOUBEN; WU, 1999)

Descrevendo a estrutura:

- **Na parte mais baixa aparecem fragmentos de informação** consideradas unidades não divisíveis (F1, F2, F3, F4.). Podem ser um parágrafo de texto, uma imagem, um vídeo *clipe* etc. O sistema não considera a parte interna de fragmentos. Fragmentos podem ser considerados unidades estáticas de texto ou podem ser gerados por programas específicos.
- **Mais acima aparecem as páginas (P1, P2..),** elementos de apresentação do material instrucional, que podem ser consideradas como uma seqüência linear

estática de fragmentos que podem ser condicionalmente incluídos (mediante a atuação do mecanismo de adaptação do sistema). Quando fragmentos são arquivos do tipo HTML, técnicas de inclusão podem ser usadas para a montagem das páginas (por exemplo, concatenação). O sistema AHA (De BRA; CALVI, 1999), utiliza páginas HTML que contêm todos os fragmentos na página e as condições (embutidas nas páginas) que determinam quando estes fragmentos serão mostrados.

- **O domínio pode também ser descrito por conceitos denominado compostos ou de contexto**, de alto nível, (C, C1, C2). As relações são definidas conforme o modelo de adaptação definido para o sistema.

Mais formalmente, no modelo do domínio tem-se conceitos e relacionamentos entre conceitos, definidos da seguinte forma:

## 1. Conceitos

- Conceitos são objetos com identificador único e uma estrutura que inclui pares de atributo-valor e âncoras de *links*. Um conceito representa um item de informação abstrato do domínio da aplicação. Ele pode ser um conceito atômico ou um conceito composto.
- Um conceito atômico corresponde a um fragmento de informação. Ele é primitivo no modelo e não pode ser adaptado. Seus componentes dependem da implementação.
- Um conceito composto tem uma seqüência de filhos (subconceitos) e uma função construtora que descreve como os filhos estão agrupados. Os filhos de um conceito composto são todos conceitos atômicos ou todos conceitos compostos.
- Páginas são conceitos compostos cujos filhos são todos conceitos atômicos. Páginas devem possuir uma estrutura que determina como devem ser apresentados os fragmentos.

**Observa-se as seguintes restrições quanto aos conceitos:** um conceito não pode conter ele mesmo, direta ou indiretamente; um conceito composto pode conter somente outros conceitos compostos ou páginas e não conceitos atômicos;

uma página contém apenas fragmentos; um fragmento pode aparecer em diferentes páginas.

## 2. Relacionamento entre conceitos.

Um relacionamento entre conceitos é uma tupla  $\langle C1, C2, T, A \rangle$ , onde  $C1$  e  $C2$  são conceitos,  $T$  é o tipo de relacionamento e  $A$  é um elemento pertencente ao domínio do tipo  $T$ . Podem ser criados relacionamentos de diversos tipos, conforme necessidades do SHA.

As primitivas de modelagem serão aplicadas no próximo item deste trabalho.

### 7.2.2 Utilização de autômatos finitos com saída na representação do modelo do domínio

No capítulo quatro, autômatos finitos determinísticos com saída foram apresentados como elementos facilitadores na construção de soluções para a área de hipermídia adaptativa.

Entre as vantagens citadas destaca-se a possibilidade de mapear a estrutura do domínio do conhecimento para um autômato com saída e com isto obter um controle da estrutura de navegação independente do material hipermídia. Cabe aqui indicar as condições do mapeamento do modelo do domínio de um SHA para autômatos com saída, de acordo com as primitivas de modelagem apresentadas no item 7.2.1, e avaliar os demais elementos da estrutura, conforme abaixo:

1. **O alfabeto de entrada do autômato:** é definido pelos nomes que identificam os relacionamentos entre conceitos (links de navegação):
  - Um relacionamento do tipo *link* é o mais comum entre dois conceitos, freqüentemente utilizado para a navegação. No caso do mapeamento para autômatos poderão ser usados os subtipos do tipo *link*, como: **próximo, anterior, pai/filho**;
  - Um relacionamento do tipo **pré-requisito de** ( $C1$  **pré-requisito de**  $C2$ ) pode ter como domínio números reais entre 0 e 1, que indicará o grau do pré-requisito, ou seja, o quanto o usuário deverá conhecer de  $C1$  antes de acessar  $C2$ .
  - Um relacionamento do tipo **inibição** significa que dados dois conceitos  $C1$  e  $C2$ , o relacionamento  **$C1$  inibe  $C2$**  significa que se  $C1$  é conhecido torna-se

indesejável ler C2. Este relacionamento diz, em outras palavras, que C1 não deve ser estudado antes de C2. Também nesse caso o domínio pode ser um número real entre 0 e 1 para indicar o grau de inibição;

- O relacionamento tipo **parte - de** representa a composição entre dois conceitos C1 e C2, indicando que C1 é parte de C2, podendo ser utilizado como domínio um valor real entre zero e 1 que indica a fração de C2 que é representada por C1.

Observa-se para os relacionamentos acima: um (sub) link deve ter pelo menos uma fonte e um destino; um conceito não pode ser pré-requisito dele mesmo; um conceito não pode inibir ele mesmo; os relacionamentos sugeridos com pesos (parte-de, pré-requisitos, inibição) sugerem, em termos de implementação, a criação do atributo peso, relativo à cada transição, mantidas a estrutura e as propriedades dos autômatos.

Outros tipos de relacionamentos podem ser criados, conforme necessidades do SHA.

2. **Alfabeto de saída:** constituído pelo conjunto de todas as formas de representação da informação que são vistas pelo usuário na tela. O tamanho e a forma de cada janela na tela, suas posições relativas, gráficos, sons, vídeo etc, caracterizam um alfabeto de saída. Isto é, o alfabeto é constituído por unidades de informação (conteúdo instrucional) representadas por páginas Web. As páginas podem ser concatenadas como uma única página Web no navegador do usuário denominada **página de saída**.
3. **Função de saída – página de saída:** a página de saída é o resultado da composição (concatenação) de diversas páginas (fragmentos) constantes do alfabeto de saída, que, no caso de SHA, podem ser condicionalmente incluídas através de uma expressão booleana ou outro recurso computacional a ser definido na implementação. Por exemplo, no caso da adaptação de conteúdo, as explicações sobre o material instrucional podem ser ou não incluídas no contexto da página de saída, dependendo de uma avaliação do modelo do usuário. As funções de saída, responsáveis pela finalização das páginas Web a serem visualizadas pelo usuário,

podem ser construídas (extensões) de forma que, controladas por algum mecanismo de adaptação, sejam capazes de gerar páginas adaptadas (adaptação de conteúdo), efetuar operações de manuseio de links (navegação adaptativa) e ainda controlar aspectos de layout (folhas de estilo). O mecanismo de adaptação seria uma das atribuições do agente de adaptação que irá compor a arquitetura geral proposta neste trabalho, a ser vista mais adiante.

A seguir, na figura 9 é mostrado um exemplo de mapeamento da estrutura de conceitos de um curso para a estrutura de autômatos (máquina de Moore).

Curso de Revisão	(R)	P0
Comandos	(1)	P1
Comandos de Seleção	(1.1)	P2
Comando if	(1.1.1)	P3
Exemplo de if	(1.1.2)	P4
Exercícios sobre if	(1.1.3)	P5
Comandos de Repetição	(1.2)	P6
Comando while	(1.2.1)	P7
Exemplo de while	(1.2.2)	P8
Exercícios de while	(1.2.3)	P9
Procedimentos	(2)	P10
Funções	(2.1)	P11
...		
...		
...		

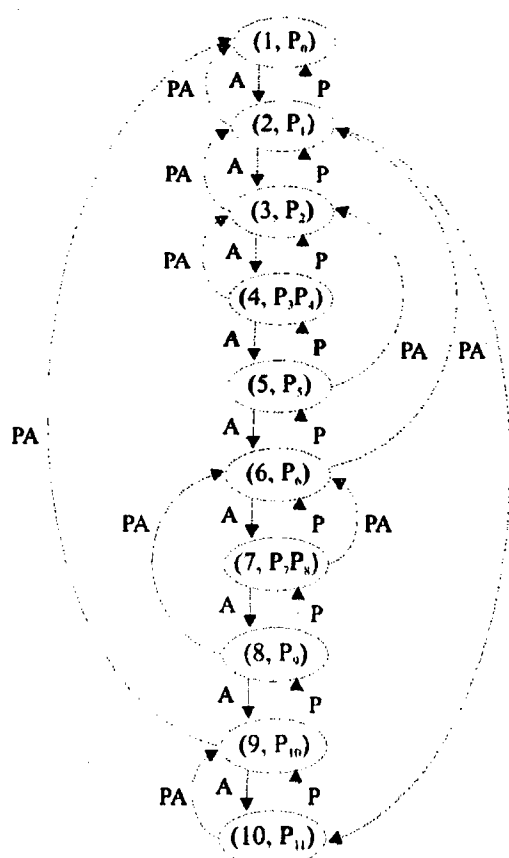


Tabela de Transição

Estado Atual	Relacionamento	Próximo Estado
1	Próximo	2
2	Próximo	3
2	Anterior	1
2	Parte de	1
2	Pre-rq-de	9
3	Próximo	4
3	Anterior	2
3	Parte de	2
4	Próximo	5
4	Anterior	3
4	Parte de	3
5	Próximo	6
5	Anterior	4
5	Parte de	3
...	...	...
...	...	...
...	...	...
...	...	...
...	...	...

Notação

P - Próximo  
 A - Anterior  
 PA - Parte de  
 PR - Pré-requisito de

Figura 9 – Autômato mapeado (máquina de Moore).

### 7.2.3 Interconexão entre o modelo do domínio e páginas hipermídia

A interconexão entre o modelo do domínio (conceitos), que organiza o conhecimento, e as páginas hipermídia que o representam perante o usuário, pode ser realizada através de diferentes métodos e depende das características de projeto do SHA. Na arquitetura proposta na abordagem deste trabalho, na qual o modelo do domínio é mapeado para autômatos, o trabalho de interconexão é realizado pela função de saída (conforme definições do capítulo



quatro) que é a responsável pela criação da página de saída a ser apresentada ao usuário em cada interação. Para realizar a interconexão podem ser aplicados os seguintes métodos (BRUSILOVSKY; PESIN, 1994):

- **Indexação por página** – o objetivo desse método é indexar todo o conteúdo de páginas hipermídia com os conceitos do modelo do domínio. Dessa maneira cada conceito está associado a uma página de índices que permite o acesso a todas as páginas que representam esse conceito.

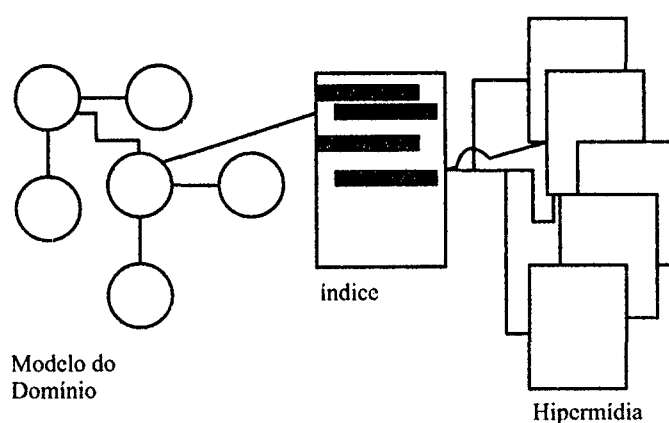


Figura 10 – Indexação por página (BRUSILOVSKY; PESIN, 1994).

- **Indexação fragmentada** – esse método é uma evolução do método anterior e permite que as páginas hipermídia sejam indexadas no nível de fragmentos, sendo assim uma indexação mais fina.

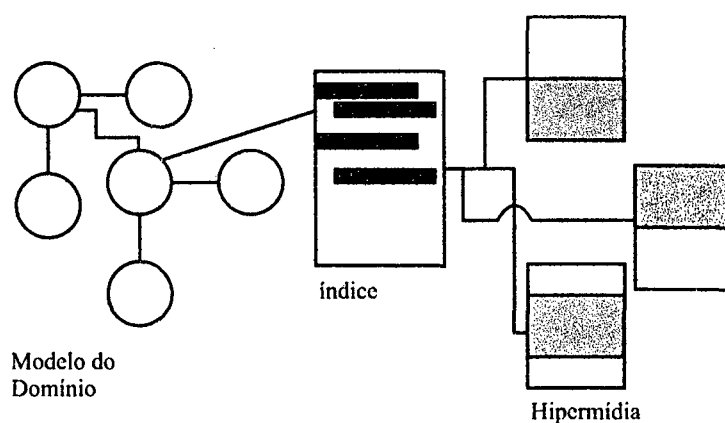


Figura 11 – Indexação fragmentada (BRUSILOVSKY; PESIN, 1994).

- **Indexação via relacionamento direto** – esse método se diferencia dos anteriores porque não utiliza página de indexação. Aqui a estrutura do hiperespaço é toda calcada sobre a estrutura do modelo do domínio, ou seja, cada conceito do modelo do domínio é representado por uma página de hipertexto e a cada relacionamento associa-se um link hipertextual.

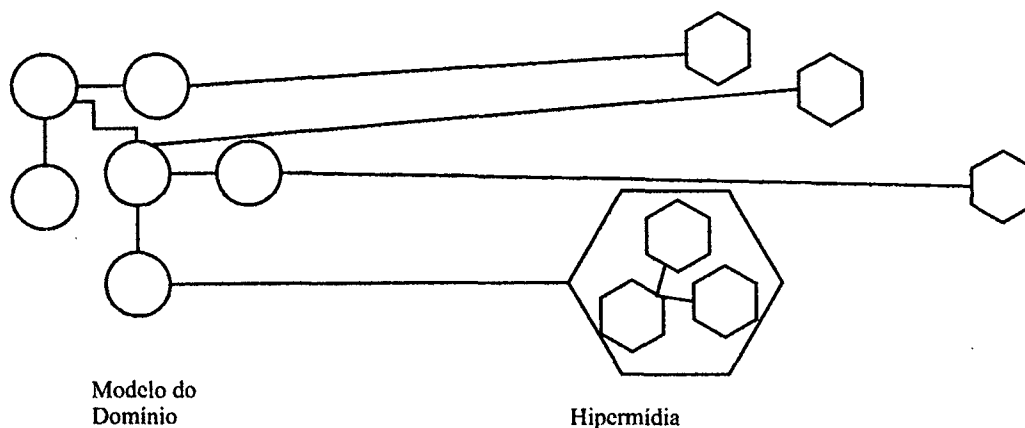


Figura 12 – Relacionamento direto (BRUSILOVSKY; PESIN, 1994).

O método de indexação via relacionamento direto é o mais simples de ser implementado e requer maior nível de fragmentação do material instrucional, o que muitas vezes é adequado, tendo-se em vista os objetivos de reaproveitamento de páginas em diversos cursos ou em diferentes seqüências de aprendizado.

### 7.3 Sociedade de agentes

Entre as definições de agentes apresentadas na revisão da literatura do capítulo cinco, foi escolhida neste capítulo a de Shoham (1997, p. 75) que define agente como: “uma entidade que funciona continuamente e de forma autônoma em um ambiente particular, freqüentemente habitado por outros agentes e processos“. A necessidade de continuidade e autonomia faz-se necessário para que o agente possa, de acordo com as mudanças encontradas no ambiente, continuar executando de forma que não precise de intervenções do usuário para cumprir suas tarefas. Por outro lado, um agente que funcione de forma contínua em um ambiente, por um longo período, seria capaz de aprender com suas experiências.

A utilização do conceito de agente na busca de soluções para sistemas distribuídos, de um modo geral, é bastante promissora e está evoluindo rapidamente. Propostas recentes de utilização de arquiteturas de sistemas multiagentes (**sociedades de agentes**) em ambientes de aprendizagem em geral, trazem muitas vantagens sobre as abordagens convencionais, nas fases de projeto, implementação e execução. Pode-se citar ganhos em flexibilidade no tratamento dos elementos que compõem o sistema, construção por módulos quase independentes (modularidade), possibilidade de dividir o problema em sub-problemas e confiança no sentido de que a resolução de um problema pode continuar ainda que haja falha em um dos agentes.

Em termos operacionais os agentes oferecem as seguintes vantagens: eficiência (tendem a consumir menos recursos de rede), menor tráfego em rede (utilizando agentes móveis), interação autônoma assíncrona (as tarefas podem ser codificadas nos próprios agentes e então serem despachadas), robustez e tolerância à falhas, conformidade com o ambiente, capacidade para trabalhar com grande volume de dados (com deslocamento do agente para servidores de banco de dados, evitando grandes volumes de transferência de dados via rede) e suporte a ambientes heterogêneos.

Segundo Giraffa (1999) a utilização do conceito de sociedade de agentes em sistemas voltados para a educação assume grande importância atualmente. Recentemente a literatura tem apresentado a noção de **agentes pedagógicos** tendo em vista que muitos sistemas com fins educacionais tem adotado o paradigma dos agentes. Agentes pedagógicos podem ser do tipo *goal driven*, dirigidos por objetivos, como, por exemplo, tutores, mentores e assistentes, e do tipo *utility-driven*, guiados pela sua utilidade no ambiente, como por exemplo, agentes da Web.

Os agentes *utility driven*, são usados como auxiliares de professores e alunos na execução de tarefas ligadas às atividades pedagógicas. Ajudam os estudantes a encontrar material didático (arquivos, diretórios, programas etc.), agendam encontros entre grupos, lembram datas de entrega de trabalhos, entre outras atividades.

Os agentes *goal driven* na Web possuem mobilidade e operam em diferentes contextos como, por exemplo, contextos textuais, hipermídia e realidade virtual. Neste caso a interação ocorre baseada no processo cooperativo ou competitivo no qual os agentes artificiais e agentes humanos se comunicam e realizam atividades.

Segundo Vassileva (apud GIRAFFA, 1999), os agentes pedagógicos podem atuar como tutores virtuais, estudantes virtuais ou companheiros virtuais de aprendizagem. As

propriedades fundamentais de agentes pedagógicos são: autonomia, habilidade social, proatividade e persistência. Além destas propriedades os agentes pedagógicos podem ser reativos, de performance contínua, serem capazes de aprender e na maioria dos casos, serem representados por um personagem.

## 7.4 Descrição geral de uma arquitetura unificada para a abordagem

A arquitetura a ser proposta na abordagem deve permitir:

- a intensificação de aspectos pedagógicos desejáveis no ambiente;
- uma estruturação do modelo do domínio que possibilite: a reutilização das páginas em diversos cursos com a eliminação da redundância; independência dos hiperdocumentos (objetos) da estrutura de navegação, cuja alteração não influencie nas páginas e vice-versa; a elaboração de sequências instrucionais com enfoques diferenciados dentro de um mesmo curso; facilidades para incluir condições de adaptabilidade;
- flexibilidade no tratamento dos elementos que compõem o sistema, construção por módulos (modularidade), expansibilidade, adaptabilidade em estruturas computacionais distribuídas e possibilidade de se dividir o problema em subproblemas facilitando o alcance da solução global.

A adaptabilidade de um SHA envolve três componentes básicos, que funcionam de forma contínua e assíncrona, que nesta abordagem serão tratados como agentes: **o agente de interface, o agente de modelagem do usuário e o agente de adaptação propriamente dito.**

O trabalho conjunto destes agentes possibilitará a realização do processo de adaptação dos SHA, conforme está descrito a seguir e ilustrado na figura 13.

Observa-se que a edição do modelo do domínio, a edição de regras de adaptação (regras pedagógicas) e a edição e avaliação de testes, serão realizados por ferramentas específicas de autoria que não estão sendo tratadas neste trabalho.

**O agente de interface** responde pelo suporte à navegação e apresentação da hipermídia adaptada ao usuário. As requisições do usuário, em relação à base hipermídia, são captadas e enviadas para o agente de adaptação para processamento e registradas em arquivos log. São

também captadas informações cadastrais dos usuários e enviadas para o agente de modelagem do usuário.

O agente de modelagem do usuário responde pela criação e manutenção da base de modelos de usuários a partir dos dados do registro de *log*, de dados cadastrais remetidos pelo agente de interface e de informações sobre a atualização do modelo do usuário enviadas pelo agente de adaptação.

O agente de adaptação processa as requisições do usuário, dispara as regras para deduzir valores de atributos que são consequência do tipo de evento ocorrido, gera a página adaptada (conteúdos e links), devolve a página para o usuário de interface e finalmente informa ao agente de modelagem do usuário quais atualizações devem ser feitas no modelo do usuário.

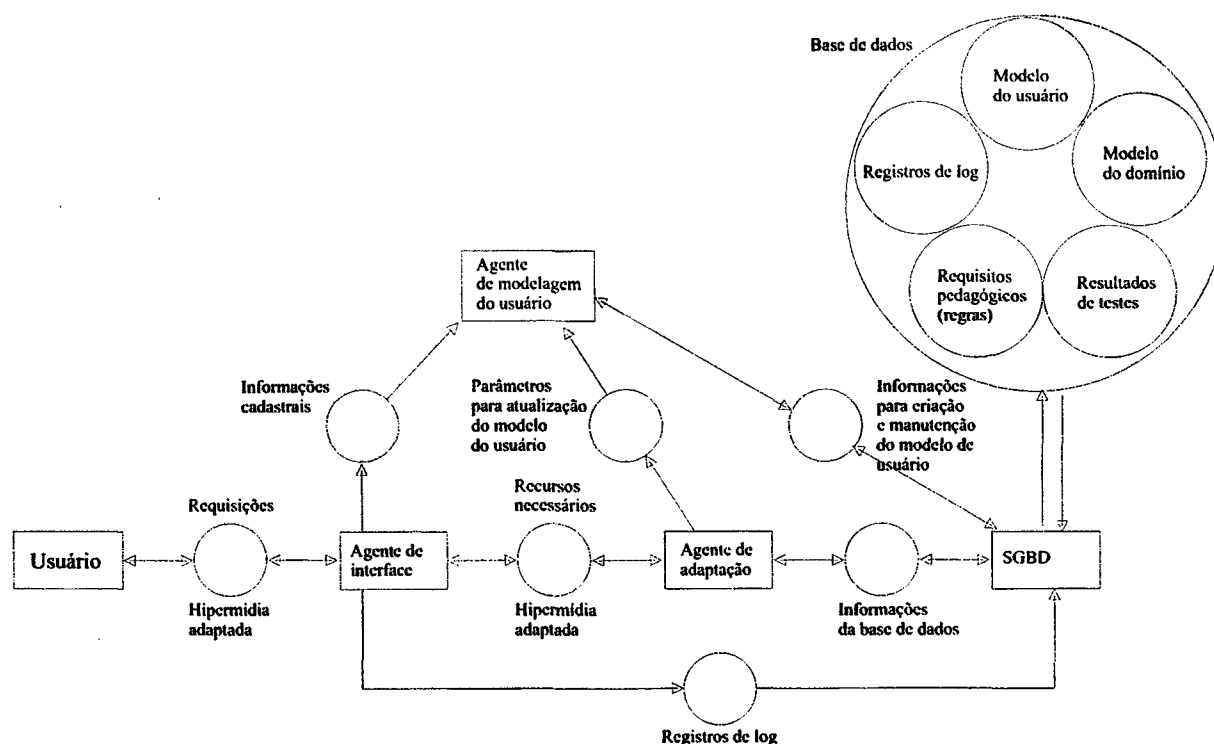


Figura 13 – Arquitetura geral proposta na abordagem (rede de estações/canais)

A interação entre os agentes ocorre da seguinte forma: agente de interface com agente de modelagem do usuário, agente de interface com agente de adaptação e agente de modelagem do usuário com agente de adaptação. Os agentes utilizarão um Sistema Gerenciador de Banco de Dados (SGBD) para armazenamento e recuperação de informações.

Os agentes identificados trabalharão em sociedade, caracterizando **um sistema multiagentes** que é uma arquitetura poderosa para a busca de soluções em sistemas de natureza distribuída, como é o caso de ambientes de aprendizagem na Web.

Com foi visto no capítulo cinco, o uso de uma arquitetura baseada em agentes permite ganhos de qualidade sob o ponto de vista pedagógico (GIRAFFA, 1999). Fatores como pró-atividade, habilidade social e flexibilidade são mais facilmente incorporadas através de agentes, atendendo de maneira mais efetiva às necessidades de um ambiente de aprendizagem.

Quanto à questão da forma de comunicação entre os agentes da sociedade, uma alternativa é uma **arquitetura baseada em troca de mensagens**, na qual os agentes comunicam-se diretamente uns com os outros enviando mensagens assíncronas, podendo opcionalmente incluir a presença de um agente facilitador de comunicação. Esse tipo de arquitetura exige que os agentes saibam os nomes e endereços uns dos outros para que as mensagens sejam devidamente encaminhadas. Esse método é mais eficiente no sentido de obter as mensagens em tempo hábil, mas por outro lado, o projeto deve prever estudos sobre a relação entre o número de agentes ativos e o número de mensagens, para evitar congestionamento no tráfego de mensagens.

Outra alternativa, que é viável e mais recomendada em casos de haver grande tráfego de mensagens, é uma **arquitetura federativa de agentes**. Esta arquitetura procura resolver o problema da comunicação interna, principalmente quando fica crítica a utilização da arquitetura anterior (*message passing*). Na arquitetura federativa os agentes da sociedade são divididos em grupos ou federações segundo um critério de agrupamento escolhido. Junto a cada grupo de agentes encontram-se os agentes facilitadores responsáveis por receber a mensagem que chega em cada grupo e encaminhá-la para o agente destinatário presente naquele grupo. A vantagem dessa arquitetura é que o agente facilitador tem a propriedade de identificar se a mensagem que chega é destinada a algum agente de seu grupo e se for o caso, fazer a devida entrega, diminuindo o tráfego na rede.

Nos itens seguintes serão discutidos mais detalhadamente os processos e estruturas envolvidas na adaptabilidade.

## 7.5 O agente de interface

O agente de interface do usuário possui as seguintes funções básicas:

- Recepção de requisições do usuário e despacho para o agente de adaptação. Essas requisições são passadas na forma de mensagens com a especificação do recurso desejado ( *link*, software, documento, mídia etc.).
- Recebimento da hipermídia adaptada do agente de adaptação e entrega da mesma ao usuário.
- Registro de informações de *log* no SGBD e envio de dados de atualização cadastral para o agente de modelagem para que o mesmo providencie a atualização da base de modelos de usuários.

## 7.6 O agente de modelagem do usuário.

O agente de modelagem do usuário tem como objetivo manter atualizados os modelos ou perfis dos usuários do sistema, para uso no processo de hipermídia adaptativa. Estes modelos são armazenados no SGBD e ficam à disposição do agente de adaptação. As principais ações do agente de modelagem são:

- a partir dos registros de log gerados pelo agente de interface (arquivo *log*) para cada usuário, é gerada a representação da rede de navegação contendo os *links* visitados.
- atualização do modelo do usuário com as informações cadastrais (dados dos usuários) enviadas pelo agente de interface;
- atualização de atributos de controle do modelo do usuário, sob solicitação do agente de adaptação. Isto ocorrerá normalmente no disparo de regras pelo agente de adaptação na ocorrência de eventos específicos, como por exemplo, na leitura de uma página ou resolução de um teste de avaliação pelo usuário.

### 7.6.1 O Modelo do Usuário – estrutura básica

Um sistema de hipermídia adaptativa precisa de uma representação interna do perfil do usuário (estado mental) para realizar a adaptação da apresentação e da navegação. O modelo do usuário representa o relacionamento entre o usuário e o modelo do domínio, guardando

informações sobre quanto o usuário conhece sobre cada um dos conceitos do modelo do domínio. A manutenção do modelo do usuário é de responsabilidade do **agente de modelagem**.

Os SHA existentes utilizam normalmente cinco características para definir o perfil do usuário: conhecimento, objetivos, história, experiência no hiperespaço e preferências.

O modelo do usuário segue a estrutura proposta para a maioria dos sistemas SHA, na qual são associados a cada conceito uma série de atributos com respectivos valores (atributos-valor). O sistema mantém para cada usuário uma instância da tabela mostrada na tabela 3 abaixo (DE BRA;HOUBEN;WU, 1999).

Conceito(id)	Conhecimento	Pág_conceito Lida	Pág_conceito pronta para ler	Outros atributos
WWW	Estudada	Falso	Falso	
WWW1	Bem conhecido	Verdade	Verdade	
WWW2	Não conhecido	Falso	Verdade	
Protocolo	Estudada	Verdade	Verdade	
Rede	Bem estudada	Verdade	Verdade	

Tabela 3 - Tabela do modelo do usuário (DE BRA;HOUBEN;WU, 1999).

O número de atributos e seus tipos e formatos dependerá de funcionalidades específicas de cada SHA. A maioria dos SHA utiliza pelos menos os seguintes atributos (DE BRA;HOUBEN;WU, 1999) :

1. **Valor do conhecimento**, ou simplesmente “conhecimento” – indica o quanto um usuário sabe sobre determinado conceito. O valor do atributo pode ser de diversos tipos como, por exemplo:
  - tipo booleano (verdadeiro ou falso) que pode ser interpretado como conceito **conhecido** ou conceito **não conhecido**;
  - um conjunto discreto de valores para cada conceito como **não conhecido**, **estudado**, **bem estudado** e **conhecido**. Um conceito torna-se **estudado** quando o usuário leu a respeito, mas não estava preparado para a leitura; ele se torna **bem estudado** quando o usuário leu e estava preparado para a leitura, e torna-se **conhecido** com o usuário foi aprovado em teste sobre o conceito;



- **um valor contínuo** numa faixa de valores reais, por exemplo, entre 0 e 1, ou percentagens entre 0 e 100. Estes valores indicam o quanto o usuário sabe sobre determinado conceito.
2. **Leitura** – indica se o usuário leu alguma página sobre o conceito. Na Web esse atributo é usado para gerar diferentes tipos de âncoras de *links*, indicando se a página já foi lida ou não. O atributo **Leitura** pode ser do tipo booleano (**lido** ou **não lido**) ou então do tipo numérico para indicar quantas vezes a página foi lida.
  3. **Pronto para a leitura** – indica se o usuário está preparado para a leitura de um determinado conceito. Esse atributo indica que os conceitos que são eventualmente pré-requisitos são já conhecidos.

As características citadas como conhecimento, história, experiência e preferências, podem ser representadas na arquitetura proposta, conforme será visto a seguir.

### 7.6.2 Avaliação do conhecimento

O conhecimento do usuário sobre os diversos conceitos do modelo do domínio é a fonte principal do processo de adaptação. O sistema deve atualizar dinamicamente o modelo do usuário, identificando a cada momento possíveis alterações do nível de conhecimento sobre determinado assunto.

Como já foi visto no capítulo três, dois métodos são normalmente utilizados nos SHA para avaliar o conhecimento do usuário: o método da sobreposição (*overlay*) baseado na estrutura de conceitos, e o método baseado em estereótipos que classifica os usuários de acordo com suas características.

O método da sobreposição é o mais utilizado por permitir uma avaliação mais fina. O método de estereótipos pode ser utilizado de forma conjunta para inicializar as variáveis do método da sobreposição. Na literatura há estudos aprofundados sobre o assunto que não pertencem ao escopo deste trabalho. Para efeito de simplificação, nessa abordagem é proposta a utilização de pelo menos dois estereótipos, classificando os usuários de acordo com o nível de conhecimento em relação ao assunto do curso e em relação a sua experiência no uso de sistemas do tipo hipermídia. Poderíamos ter, por exemplo, usuários nas classes: iniciante,

intermediária e avançada. Um aluno poderia ser classificado na classe iniciante em relação ao assunto do curso e na classe intermediária em relação à experiência em ambientes hipermídia.

No **método de sobreposição**, para determinar o conhecimento do usuário em relação a conceitos do modelo do domínio, podem ser utilizadas diferentes abordagens que tem como base o monitoramento da navegação realizada pelo usuário, verificando quais as páginas visitadas e o seu desempenho em testes de avaliação e nas demais atividades pedagógicas.

Uma **abordagem interessante trabalha com fatores de confiança** (ANJANEYULU, 1996). Para um determinado usuário, o atributo **conhecimento** será representado por um coeficiente de confiança (CF) que indicará o quanto o usuário sabe sobre o conceito, usando faixa de valores entre -1 (usuário não conhece o conceito) e 1 (usuário sabe o conceito). O valor inicial de CF é zero.

O professor autor deverá, na fase de implantação do curso, indicar para cada teste/ou atividade previstos no esquema pedagógico, quais conceitos da estrutura do domínio correspondente serão avaliados, e o fator de confiança na avaliação (tem-se uma relação entre testes e conceitos). Por exemplo, se o aluno vai bem no teste A, qual é o fator de confiança em afirmar que o aluno conhece determinado conceito B? Ou ainda, inversamente, se o aluno vai mal no teste A, qual é o fator de confiança em afirmar que o aluno não conhece o conceito B? Para apurar valores, tem-se então dois fatores de confiança: **CF\_sabe** indica a confiança em afirmar que o usuário conhece o conceito, e **CF\_nãosabe** indica a confiança em afirmar que o usuário não conhece o conceito. Estes dois coeficientes serão usados na composição do cálculo do coeficiente geral CF, e possuem valores na faixa de 0 a 1.

O sistema, baseado em técnicas desenvolvidas no Modelo *Mycin* (Sistema Especialista), usa uma das seguintes fórmulas para calcular o novo CF (CFnovo) para um conceito, baseado no antigo CF (CFvelho), sendo CFq dado por CF\_sabe ou por CF\_nãosabe. Se o estudante acerta a questão o CF será CF\_sabe e se erra a questão CF será -CF\_nãosabe.

$$CF_{novo} = CF_{velho} + CF_q (1 - CF_{velho}) \text{ quando } (CF_{velho}, CF_q > 0) \quad (1)$$

$$CF_{novo} = CF_{velho} + CF_q (1 + CF_{velho}) \text{ quando } (CF_{velho}, CF_q < 0) \quad (2)$$

$$CF_{novo} = (CF_{velho} + CF_q) / (1 - \min(|CF_{velho}|, |CF_q|)) \text{ caso contrario } (3)$$

Este modelo incrementará o CF cada vez que o usuário acertar uma questão e diminuirá cada vez que o usuário errar. Pode-se considerar, por exemplo, que o aluno conhece um

determinado conceito, se o CF correspondente, por exemplo, estiver acima de 0,6. Adicionalmente a lógica *fuzzy* pode ser utilizada para uma abstração das faixas de variação.

Teste	Conceito	CFsabe	CFnsabe
1	01	0,2	0,2
2	01	0,6	0,6

CF - (-1,1) - início CF = 0

Aluno	Conceito	CF
0001	01	0,6
0001	02	1,0
0002	01	0,4

Utilizando coeficientes de segurança

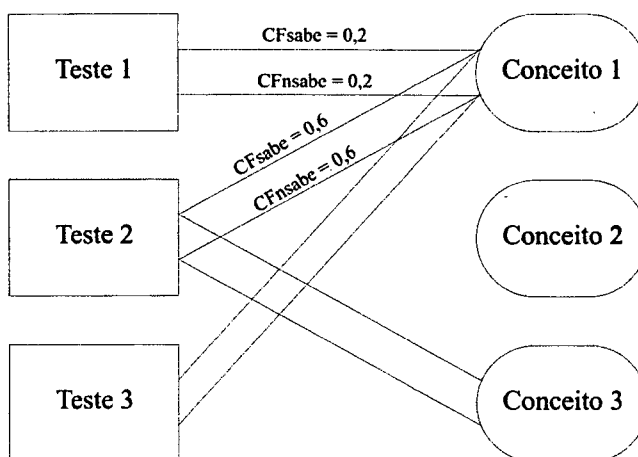


Figura 14 – Avaliação do conhecimento através de coeficientes de confiança.

O mesmo procedimento pode ser feito utilizando a fórmula de Bayes para cálculo de probabilidade na relação entre eventos dependentes: se A ocorre qual a probabilidade de B ocorrer, e vice-versa. As técnicas citadas podem ser utilizadas também em relação a objetivos de aprendizagem.

**Uma segunda abordagem** é a de usar o conceito de Redes Bayesianas, que estende o teorema de Bayes para arquitetura de redes, com propagação para os nós da rede. A utilização plena dessa técnica requer a determinação das probabilidades de ocorrência (condições) entre os conceitos de toda a rede. Essa técnica tem sido utilizada em alguns sistemas adaptativos, como por exemplo, KBS- Hiperboock, já citado na revisão bibliográfica. Como vantagem principal dessa técnica temos uma precisão maior no processo de determinação do conhecimento do aluno, pois ocorre, a cada instância da aplicação do método, a propagação na rede das probabilidades determinadas. Como desvantagem, existe a questão da autoria e da possível sobrecarga de processamento em redes maiores. O professor autor deverá definir as relações de probabilidade condicional entre os nós da rede, processo esse que pode ser complicado e extenso para determinados tipos de cursos. Além disso, é preciso considerar que

num ambiente de aprendizagem poderá ocorrer diversidade de autores com formações diferentes, não familiarizados com o processo.

Uma terceira abordagem parte do princípio que o relacionamento entre conceitos na hierarquia é definida em termos percentuais pelo usuário autor da aplicação hipermídia, ou seja, fica definido o quanto o conhecimento de um determinado conceito contribui para o conhecimento de um conceito relacionado. Por exemplo, na figura abaixo se tem que as páginas P1 e P2 contribuem em 50% cada para o conhecimento de C1 e C1 contribui em 40% para o conhecimento de C. Caso a página P1 seja lida, uma regra de adaptação seta o conhecimento de P1 para 100% e dispara uma outra regra (outra instância) que atualiza o conhecimento de C1 para 50%. Em seguida, esta última regra dispara mais uma regra (outra instância) que seta o conhecimento de C em 20% (ou 40% de 50%). Neste caso o mecanismo de inferência deve permitir que a execução de uma regra possa disparar outra regra.

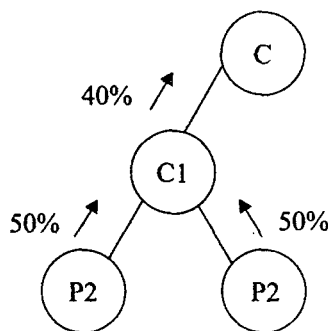


Figura 15 – Propagação do conhecimento em estrutura de conceitos

### 7.6.3 Objetivos do usuário, história e preferências.

Os objetivos do usuário estão relacionados com o contexto da aprendizagem. Os objetivos dos usuários são voláteis, ou seja, podem mudar freqüentemente. Uma representação possível dos objetivos usa uma árvore de atividades. Outra representação possível é utilizar um conjunto de pares (objetivo-valor), onde valor é a probabilidade deste objetivo ser o objetivo corrente do usuário.

A **história** do usuário em tópicos relevantes, não relacionados propriamente com os assuntos do sistema hipermídia, é fator importante na produção do modelo do usuário. Por exemplo, conhecimentos sobre a linguagem Pascal podem ser considerados como relevantes para um usuário que faz um curso de Java num ambiente hipermídia.

A **experiência** refere-se à experiência do usuário com sistemas baseados em hipertexto ou hipermídia, adaptativos ou não.

**Preferências** - os usuários de sistemas de hipermídia podem ter diversas preferências, por exemplo, fontes, cores etc. Essas características devem ser obtidas junto aos usuários. Sistemas de hipermídia adaptativa devem ser capazes de oferecer diversas alternativas de preferências possibilitando que os próprios usuários façam as adaptações de acordo com suas preferências.

As características citadas acima também podem ser representadas na arquitetura proposta relacionando os conceitos com pares atributo-valor.

## 7.7 O agente de adaptação

O agente de adaptação, nessa abordagem, é o motor central de um sistema de hipermídia adaptativa. Em termos gerais é o responsável pela adaptação de conteúdo, adaptação de *links* e atualização do modelo do usuário (via agente de modelagem do usuário) através do uso de mecanismos de inferência.

As principais funções executadas por esse agente são:

1. Recepção de requisições de usuários, via agente de interface.
2. Execução da adaptação, tendo como base as requisições recebidas, o modelo do usuário, o modelo do domínio (com mapeamento para autômato com saída) e regras de adaptação (ou regras pedagógicas). Isto significa selecionar páginas, selecionar fragmentos e construir páginas. A adaptação envolve ainda a adaptação de *links* dependendo do estado do link (habilitado, escondido etc.). É muito comum nos SHA existentes o uso de construtores de páginas embutidos no código HTML, na forma de scripts contendo regras de adaptação específicas. Este procedimento não favorece a idéia de reutilização das páginas em diferentes cursos. Na arquitetura proposta neste trabalho os construtores, são representados por funções de saída estendidas vinculadas aos autômatos, que estarão embutidos

no agente de adaptação, ou então serão agentes colaboradores que poderão ser construídos com finalidades específicas, aproveitando as características da estrutura multiagente e do modelo do domínio proposto.

3. Atualização do modelo do usuário cada vez que o usuário visitar uma página ou então resolver um exercício. São atualizados os atributos do modelo do usuário, dependendo de regras de adaptação. **Para atualizar o modelo do usuário** durante as interações, o agente de adaptação passa parâmetros para o agente de modelagem do usuário que faz a respectiva atualização.
4. Envio do material hipermídia adaptado para o processo de interface.

As atividades típicas realizadas quando ocorre um evento de requisição de uma página, por um determinado usuário (aberta uma seção e identificado um usuário) são:

1. O agente de adaptação recupera da base modelo do usuário, atributos dos diversos conceitos.
2. Outros atributos que não pertencem ao modelo do usuário, mas que constam nas regras, são inicializados com valor *default* (evita-se com isto grande número de regras). Se o atributo **pronto para ler** não é usado no modelo do usuário e é utilizado nas regras, pode-se setar este atributo como **verdade** para todos os conceitos, como valor *default*, e então definir regras para muda-lo para **falso** quando necessário. **Observa-se** que as atualizações no modelo do usuário somente podem ser feitas pelo agente de modelagem do usuário, cabendo ao agente de adaptação enviar parâmetros de atualização para o referido agente.
3. O valor do atributo conhecimento do conceito associado com a página requisitada será incrementado e o atributo **leitura** também muda para **verdade** (via passagem de parâmetros para o agente de modelagem do usuário, após disparo da regra adequada).
4. Regras pedagógicas são aplicadas para deduzir valores que são consequência do evento que ocorreu. Pode haver propagação de regras e neste caso é importante que a definição das mesmas seja feita de maneira que não ocorram **loops** infinitos e que as atualizações sejam independentes da ordem de aplicação das regras. **Eventuais atualizações** que serão necessárias no modelo do usuário, em decorrência da aplicação das regras, serão enviadas em forma de parâmetros para o agente de modelagem de usuário que tomará as devidas providências.

5. Utilizando valores atualizados dos atributos do modelo do usuário, um construtor de páginas é acionado para gerar a apresentação da página requisitada (considerando condições de inclusão de fragmentos); *links* são anotados ou escondidos. A página gerada é devolvida para o agente de interface.

Como pode ser visto, em cada evento é acionado o agente de adaptação, o modelo do usuário é lido, a página é gerada e o modelo do usuário atualizado é salvo.

Em conformidade com as funções que executa, o agente de adaptação possui:

- Um modelo de ensino (modelo de adaptação), baseado num conjunto de regras de produção (regras pedagógicas) gerais e específicas, que definem como deve ocorrer a adaptação de conteúdo e a adaptação de links de navegação;
- um mecanismo de adaptação propriamente dito, capaz de efetuar a geração de páginas e a manipulação de links, de acordo com as regras pedagógicas definidas, tendo como base os atributos do modelo do usuário. O mecanismo de adaptação prove também a atualização do modelo do usuário cada vez que o usuário visita uma página ou resolve um teste, disparando as regras adequadas após a geração. O mecanismo de adaptação tem controle e acesso total às estruturas do modelo do domínio mapeado para autômatos, via SGBD. As funções de saída relativas aos autômatos são na realidade as funções construtoras das páginas de saídas adaptadas.

O **mecanismo de inferência** proposto para o agente de adaptação é baseado em regras de produção que fazem a conexão entre o modelo do domínio, o modelo do usuário e a apresentação a ser gerada.

As regras são tipicamente do tipo “if <condição> then <ação>”, onde a condição especifica a ocorrência de um evento externo, como “página acessada”, e pode conter expressões booleanas referindo-se a valores de atributos do modelo do domínio ou do modelo do usuário. A ação pode ser a atualização de um atributo do modelo do usuário, ou pode especificar que um fragmento deve ser mostrado, ou que um link seja ativado, etc.

De Bra, Wu e Houben (1999), propõem dois tipos de regras pedagógicas: regras gerais, normalmente implícitas no SHA e regras específicas definidas pelo usuário autor, descritas a seguir.

**Uma regra geral** é uma tupla <R, PH, PR> onde:

- R indica que é uma regra tipo *trigger*;

- PH é a fase para a sua execução - tem o valor *pré* se executada antes ou durante a apresentação da página ou páginas, ou *post se* executada depois da apresentação;
- PR é o campo propagação indica se a regra pode disparar outras regras.

A sintaxe depende do SHA considerado. Uma regra geral usa e possibilita alterações em variáveis que denotam identificadores de conceitos, atributos, âncoras, partes de especificações de apresentações e atributos do modelo do usuário para conceitos e relações entre conceitos.

**Uma regra específica** é uma tupla  $\langle R, SC, PH, PR \rangle$ , onde :

- R indica que é uma regra do tipo *trigger*;
- SC é um conjunto de componentes de conceitos utilizados na regra;
- PH é a fase para a sua execução - tem o valor *pré* se executada antes ou durante a apresentação da página ou páginas, ou *post se* executada depois da apresentação;
- PR é campo de propagação (booleano). A regra usa e seta atributos do modelo do usuário e predicados sobre determinados conceitos de SC.

Regras específicas tem prioridade sobre regras gerais e são usadas para definir exceções das regras gerais.

Para o controle de execução, as regras são organizadas em 4 grupos de acordo com as fases do processo de adaptação e para cada grupo é proposto um algoritmo de controle de execução das regras:

- **Grupo 1 (fase pré)** - regras que respondem pela inicialização de atributos voláteis do modelo do usuário: um algoritmo executa regras de inicialização de atributos voláteis do modelo do usuário;
- **Grupo 2 (fase pré)** – regras que selecionam páginas para mostrar na tela e fragmentos para incluir nessas páginas e atualizam o modelo do usuário antes da geração das páginas;
- **Grupo 3 (fase Pré)** - regras para preparar a apresentação ( o que vai ser mostrado);
- **Grupo 4 (fase post)** – regras para atualizar o modelo do usuário.

Como exemplo, uma regra típica para a **fase Pré**, seria aquela que desabilita um link para um conceito cujo pré-requisito não é ainda conhecido. Um regra típica para a **fase Post** seria aquela que muda o atributo conhecimento de determinado conceito para “bem conhecido” se o conceito (página) foi acessado e todos pré-requisitos estavam atendidos.



Observa-se que o esquema de regras proposto utiliza regras do tipo CA (condição-ação). Outra possibilidade seria a utilização de regras do tipo ECA (evento-condição-ação).

Numa primeira visão tende-se a usar ECA, pois o SHA estará sempre reagindo a eventos gerados pelo usuário quando clica numa ancora de um link. No entanto após esse evento inicial, diversas regras poderão ser disparadas com mudanças no modelo do usuário. Propõe-se o uso de regras do tipo CA pela facilidade de controle. O fato de a condição ser verdadeira constitui o evento para disparar a regra.

A seguir serão mostrados alguns exemplos de regras. Não há no caso, preocupação com a sintaxe das regras, pois cada SHA utiliza notação própria.

**Exemplo 1:** Manutenção do modelo do usuário.

< Acessado (C)  $\rightarrow$  C.read:= verdade, post, verdade >

Essa regra indica que quando um usuário acessa uma página, o atributo “leitura”, no modelo do usuário, do conceito correspondente é mudado para verdadeiro.

Essa regra é executada na fase pós – o algoritmo controlador disparará outras regras do mesmo grupo que tenham leitura na parte esquerda da regra.

**Exemplo 2:** Manutenção do modelo do usuário.

< Acessado (C) e C.estudado = true  $\rightarrow$  C. conhecimento: = bem\_estudado, pré, verdade>

Essa regra, através da função “Acessado” indica que quando uma página é “relevante” e é acessada, o valor do conhecimento correspondente no modelo do usuário é atualizado para “bem\_estudado”.

**Exemplo 3:** Manutenção do modelo do usuário.

< teste (C) e C.teste > 70  $\rightarrow$  C.conhecimento: = conhecido, pré >.

Essa regra, através da função “teste” indica que quando o usuário faz um teste relativo ao conceito C, e o resultado foi maior que 70, seu conhecimento sobre o conceito C é atualizado para “conhecido”.

**Exemplo 4:** Visibilidade de fragmentos quando uma página é acessada.

< Acessado (C) e  $F \in C.filhos$  e F. pronto para leitura = verdade  $\rightarrow$  F.apres:= mostre, pré, false>

Essa regra indica que acessada a página correspondente ao conceito C serão mostrados os fragmentos desta página que estiverem liberados para a leitura.

**Exemplo 5:** Pré-requisitos e inibição

< C1.conhecimento < 0,8 e C1 é pré-requisito de C2  $\rightarrow$  C2.pronto para leitura := falso , pré>

Essa regra indica se C1 não é conhecido e é pré-requisito de C2 então C2 não está pronto para a leitura.

$\langle C1.conhecimento > 0,8 \text{ e } C1 \text{ é inibidor de } C2 \rightarrow C2.pronto \text{ para leitura:} = \text{falso, pré} \rangle$

Essa regra indica que se C1 é conhecido e é inibidor de C2 então C2 não está pronto para a leitura.

$\langle C1 \text{ é pré-requisito de } C2 \text{ e } C2 \text{ é pré-requisito de } C3 \text{ e } C1.não \text{ conhecido} \rightarrow C3:pronto \text{ para leitura:} = \text{falso, pré} \rangle$

Note-se o efeito de propagação na cadeia de pré-requisitos.

#### **Exemplo 6:** Manipulação de *links* ( anotação)

$\langle A1 \text{ é ancora de } C1 \text{ para } C2 \text{ e } C2 = \text{pronto para a leitura e } C2.leitura = \text{falso} \rightarrow A1.cor := \text{verde, pré} \rangle$

Essa regra indica que se A1 é ancora de C1 para C2 e C2 não foi lido, então a âncora de A1 muda para verde.

Com relação ao processo de inferência utilizando regras de produção, é preciso considerar a possibilidade de haver conflitos em regras, ou seja, duas ou mais regras diferentes conduzindo à mesma conclusão. Nestes casos existe a possibilidade de se eliminar conflitos através de diversos mecanismos. Um deles é o de estabelecer prioridades de execução. No caso deste modelo é recomendado o seguinte esquema de prioridades: regras do grupo 1; regras do grupo 2; regras dos grupos 3 e 4 (DE BRA;HOUBEN;WU, 1999).

## **7.8 Arquitetura Computacional para o SHA**

Para suportar a arquitetura proposta na abordagem, é feito a seguir um estudo e a definição de uma arquitetura computacional do tipo cliente servidor em três camadas.

### **7.8.1 Arquitetura Cliente Servidor em três camadas**

Na arquitetura cliente/servidor, surgida no início da década de 90, a principal característica é a interoperabilidade e portabilidade, ou computação baseada em padrões, que possibilita a computadores, redes e softwares, provenientes de fabricantes diferentes, interagir e dessa forma trabalhar em harmonia sem maiores problemas de configuração.

Esta arquitetura pode ser vista como uma evolução dos modelos utilizados anteriormente: os sistemas centralizados e os micros isolados.

A principal vantagem dessa arquitetura é permitir a separação das aplicações do processamento de transações. Nesses aplicativos distribuídos, a porção servidora é responsável pela segurança e pela integridade dos dados, enquanto que a porção cliente serve para exibição dos dados utilizando uma interface amigável.

Atualmente são utilizados dois modelos de arquitetura cliente servidor: o modelo duas camadas (*2-tier*) e o modelo três camadas (*3-tier*).

No **modelo duas camadas**, o cliente conversa diretamente com o servidor. As regras de processamento podem estar em qualquer uma das pontas, conforme figura abaixo.

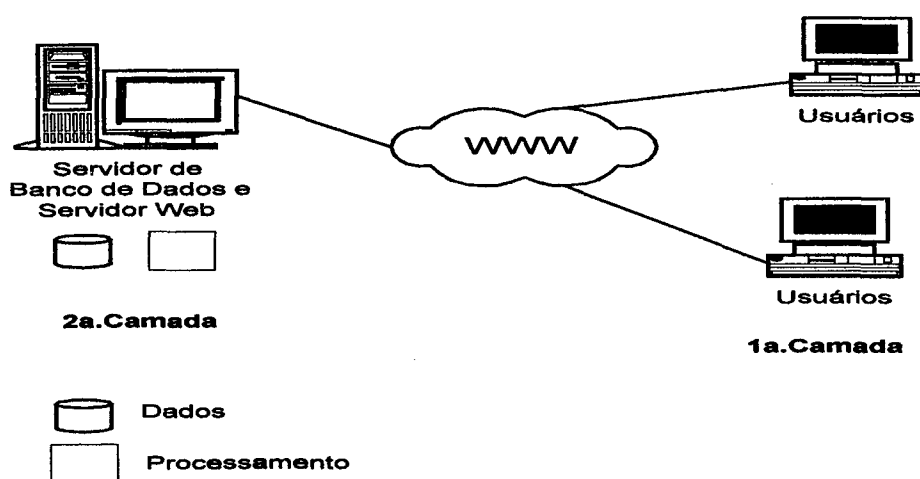


Figura 16 – Arquitetura cliente servidor com 2 camadas

Pode-se conceituar as camadas como segue (LEAL; NUNES APUD PEREIRA, 1999):

**1ª Camada:** a estação cliente – é uma junção entre um usuário e um programa de computador. Na arquitetura em duas camadas o usuário tem a interface do sistema exclusivamente para ele.

**2ª Camada:** o servidor de banco de dados e servidor *Web* - é onde ocorrem o processamento das transações de banco de dados e o processo de gerenciamento de páginas.

Com isso algumas vantagens são observadas:

- a integridade dos dados, pois somente é possível acessar os dados através da aplicação;

- o desempenho é dependente do servidor e não do cliente. O alto desempenho deve estar nos servidores e não é necessário nas estações cliente;
- tráfego reduzido na rede: somente os dados requisitados pelo cliente trafegam pela rede.

Apesar disso, o modelo de duas camadas tem desvantagens, que são: a administração de modificações e melhorias requer atualizações em várias estações clientes, ou seja, se ocorre uma alteração no sistema, terá que ser feita a atualização nos clientes; dificuldades em compartilhar processos; dificuldades com segurança.

Um grande problema causado pela descentralização do desenvolvimento dos aplicativos é a redundância de códigos e regras de processamento. Duas das soluções para este problema são a orientação a objetos e a arquitetura em três camadas.

Segundo Pereira (1999), a **arquitetura em três camadas** é uma implementação sofisticada, nas quais as regras de processamento são manipuladas por um agente, cujo papel é servir de intérprete entre as duas pontas, mapear solicitações feitas a múltiplos servidores e coletar e sintetizar respostas, ou seja, uma camada intermediária é criada entre o banco de dados e a aplicação cliente.

Uma das principais funções desta arquitetura é armazenar as regras de negócio e a lógica da aplicação (separação entre uso e apresentação, regras de processamento e dados). No lado cliente, a aplicação fica responsável apenas pelo controle da interface com o usuário. Qualquer alteração feita nessa camada será imediatamente assumida por todas as aplicações e pelo próprio banco de dados. Desse modo, o tempo e o esforço gastos para implementar uma alteração é reduzido significativamente. A figura 17 abaixo ilustra a arquitetura em três camadas.

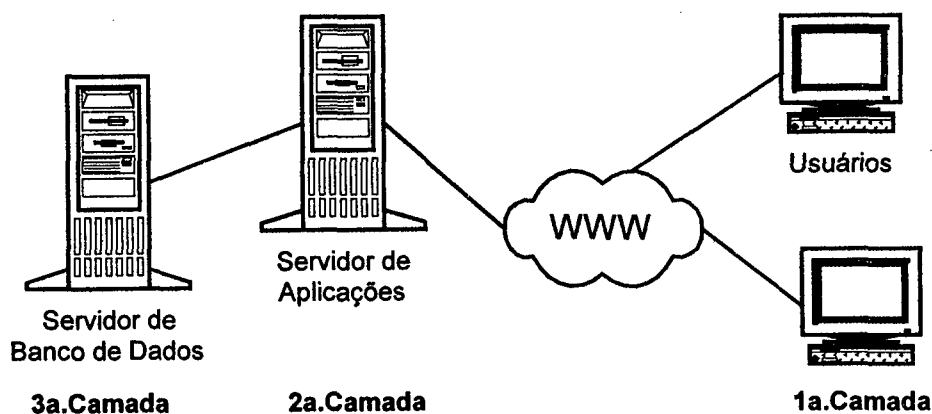


Figura 17 – Arquitetura cliente servidor com 3 camadas (PEREIRA, 1999).

Na **1ª camada**, também denominada cliente, encontram-se os objetos responsáveis pela interface, que estarão atuando sobre uma página HTML, executada em um *browser*.

Na **2ª camada**, estão as regras do negócio, que de fato processam os dados. Essa camada é composta por um servidor conhecido como Servidor de Aplicações. Esse servidor é responsável pelas regras e pelo gerenciamento do fluxo de dados.

Na **3ª camada**, está o Servidor do Sistema Gerenciador do Banco de Dados, que armazena os dados que serão requisitados pelo Servidor de Aplicações.

### 7.8.2 Arquitetura Computacional Proposta para o SHA

A arquitetura cliente/servidor em 3 camadas atende perfeitamente aos requisitos da arquitetura proposta para os SHA. Ela permite uma visualização mais clara do fluxo de comunicação entre seus componentes além das outras vantagens citadas anteriormente. Isso é bastante relevante, quando o que se pretende é a otimização na execução das aplicações que rodam sobre essa plataforma. O ambiente JAVA é recomendado (não é condição necessária) por possuir as características de portabilidade e por possuir as ferramentas de software necessárias para o ambiente.

A figura 18 mostra a arquitetura para o modelo proposto e o fluxo de comunicação entre os componentes projetados.

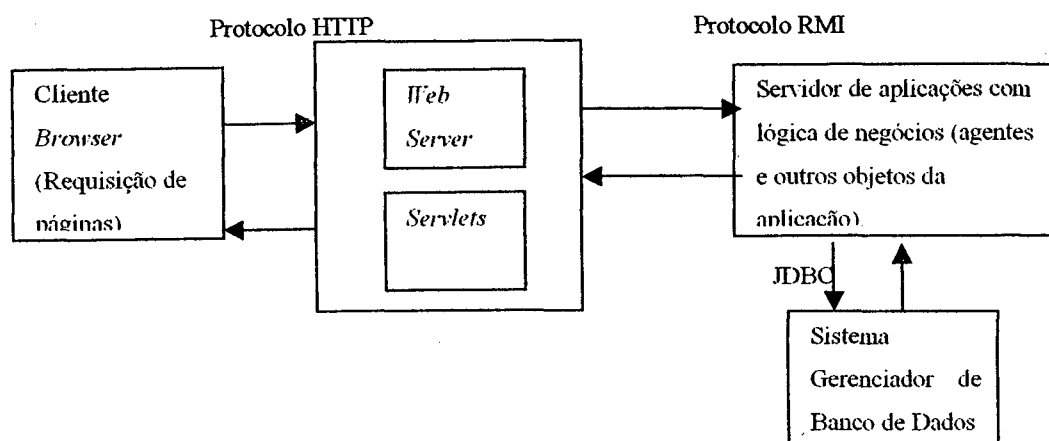


Figura 18 - Arquitetura computacional para SHA, em 3 camadas utilizando ambiente JAVA

Através de um acesso às páginas da aplicação, o usuário solicita e/ou envia informações, via protocolo HTTP, ao servidor *Web*. Esse, por sua vez, faz a chamada aos *Servlets* para o processamento das requisições. Os *servlets* fazem as invocações necessárias aos métodos dos objetos (agentes) instalados no servidor de aplicações (via RMI) para que a solicitação requisitada seja cumprida. Respondendo à invocação dos métodos, o Servidor de Aplicações faz a conexão com o Servidor de Banco de Dados, utilizando o *driver* JDBC (*Java Database Connectivity*).

O **JDBC** é um conjunto de classes e interfaces escritas em Java que permite que outros programas escritos em Java enviem comandos do tipo SQL (*Structured Query Language*) para o processamento em um sistema gerenciador de banco de dados relacional. Em outras palavras, JDBC é uma API (*Application Program Interface*) que representa uma abordagem robusta e orientada a objetos para acesso a banco de dados relacionais.

**Servlets** são classes que utilizam recursos suplementares da API Java para estender as funções do servidor WEB. Uma vantagem importante é que após atender uma requisição eles permanecem ativos (qualidade de persistência). Outras vantagens que oferecem os *servlets* podem ser traduzidas por: portabilidade para a maioria dos sistemas operacionais e servidores Web; capacidade de utilizar toda a potencialidade das API's Java; eficiência nas chamadas pois um *servlet* carregado permanece na memória do servidor como uma única instância do

objeto; segurança, por serem escritos em Java (herdam as propriedades da linguagem); integração fácil com o servidor permitindo a cooperação na execução de tarefas; extensão, pois a API *servlet* foi criada para ser facilmente extensível..

A arquitetura projetada permite o processamento distribuído através da rede. Tradicionalmente, a comunicação remota entre objetos é realizada através de *sockets* e um protocolo personalizado. No ambiente Java há dois mecanismos adicionais para a comunicação remota objeto-a-objeto: uma interface na arquitetura de distribuição de objetos **CORBA (Common Object Request Broker Architecture)** e a **RMI (Remote Method Invocation – específica para Java)**. Esses mecanismos permitem de maneira simples que um objeto Java invoque um método em outro objeto Java através da rede.

A interface RMI permite que objetos Java, em computadores diferentes, comuniquem-se entre si. Cada objeto remoto implementa uma interface que especifica quais de seus métodos podem ser invocados pelos clientes. Os clientes podem invocar métodos de um objeto remoto quase exatamente da mesma maneira que eles invocam métodos locais.

Uma vantagem, muito importante da RMI é o fato de ser possível o deslocamento de dados e estados de uma classe (comportamento móvel) de um cliente para o servidor ou vice-versa, com a utilização de implementações utilizando Java. Sendo assim, é perfeitamente possível construir uma **arquitetura baseada em agentes**.

A combinação Java/RMI tem sido largamente utilizada na construção de agentes inteligentes em função de sua capacidade de execução em diferentes plataformas. Isso inclui a construção de agentes de interface e agentes móveis capazes de se mover pela rede e executar tarefas em diversos computadores.

No escopo deste trabalho, o sistema de hipermídia protótipo (ver Apêndice A), foi construído usando a arquitetura computacional aqui proposta, utilizando a combinação Java/RMI.

## 7.9 Visão geral sobre o sistema de hipermídia adaptativa protótipo

### 7.9.1 Introdução

O Sistema de Hipermídia Adaptativa Protótipo, denominado Uniclick2000, foi projetado para ser um ambiente de aprendizagem na Web com a finalidade de apoiar o ensino à

distância e o ensino presencial. O projeto original do Uniclick2000 foi dividido em duas grandes etapas, normalmente recomendadas por pesquisadores da área de hipermídia adaptativa: para a primeira etapa foi prevista a modelagem e a implementação de um sistema hipermídia convencional (páginas estáticas); para a segunda etapa foi prevista a modelagem e a implementação dos módulos de adaptação. A preocupação básica na primeira etapa foi de projetar o modelo do domínio com mapeamento para autômatos com saída, com capacidade para suportar os requisitos dos mecanismos de adaptação previstos para a segunda etapa. Outro aspecto importante foi de escolher uma arquitetura computacional adequada para suportar o Uniclick2000 como um todo, no caso a arquitetura cliente servidor em três camadas, conforme sugerido no capítulo sete. No escopo desta Tese o trabalho ficou restrito à primeira etapa.

### **7.9.2 Objetivos**

O desenvolvimento de um sistema protótipo no escopo deste trabalho de Tese, teve com finalidade testar e validar as tecnologias para sistemas distribuídos na Web bem como as estruturas e processos da arquitetura proposta, principalmente a utilização de autômatos como elementos centrais no controle da base hipermídia.

### **7.9.3 Características gerais do sistema Uniclick 2000**

As principais características do sistema protótipo Uniclick2000 são:

- arquitetura tecnológica do tipo cliente servidor em três camadas, conforme proposição feita no contexto deste trabalho;
- desenvolvimento utilizando técnicas de orientação a objetos;
- modelo do domínio – baseado no currículo seqüencial, com estrutura de navegação independente das páginas através da utilização de autômatos finitos com saída. A utilização da estrutura de navegação independente das páginas favorece diversos aspectos de aplicações hipermídia já comentados neste trabalho, incluindo facilidades na manipulação de links;
- a interface gráfica é integrada e oferece facilidades ao professor na criação de cursos e aos alunos no acesso ao material instrucional;



- a cada curso podem ser associados um ou mais autômatos, o que possibilita a criação de roteiros diferenciados, conforme estratégia pedagógica adotada, aproveitando páginas já criadas. Por exemplo, curso introdutório, curso avançado, revisão de assuntos, preparação para prova. O aluno poderá selecionar qual o material instrucional que deseja estudar (mais tarde o próprio sistema vai recomendar o material instrucional conforme o modelo do aluno);
- o sistema oferece ferramentas para avaliação dos alunos através de: testes com correção automática em diversas modalidades (múltipla escolha, verdadeiro ou falso, associação de colunas); atividades para correção pelo professor e participação em Fórum;
- o professor pode planejar o esquema de avaliação, ponderando as atividades a serem desenvolvidas. O sistema avalia a cada instante resultados obtidos disponibilizando informações ao professor e aos alunos;
- são oferecidas as seguintes ferramentas de comunicação: Fórum, Chat, Perfil, Tutoria (tira dúvidas com o professor), Galeria (publicação de trabalhos de alunos), FAC (dúvidas mais freqüentes) e Mural. As ferramentas Fórum e FAC são objetos que podem ser inseridos na estrutura de nós hipermídia, a critério do professor, constituindo-se em elementos específicos no contexto dos conceitos envolvidos;
- é oferecida a ferramenta Mediateca, na qual o professor pode armazenar material de interesse dos alunos;
- o modelo permite a criação de diversos tipos de usuários, fornecendo níveis de autorização de acesso diferentes. Inicialmente estão previstos quatro tipos de usuários:
  - a. **professor autor** – o seu trabalho vai desde a concepção de um curso até a sua implantação no ambiente. Tarefas típicas: pré-autoria do curso, isto é, concepção e preparação do material instrucional, incluindo definição de objetivos de aprendizagem, textos, exemplos, exercícios e atividades; autoria do curso envolvendo a sua implantação no ambiente, criação do modelo do domínio no sistema e definição de plano de avaliação geral para o curso;
  - b. **professor tutor** – é o professor responsável pela execução do curso. Tarefas típicas: suporte aos alunos via ferramentas Tutoria, Fórum, Chat e

Mensagens; correção das atividades desenvolvidas pelos estudantes e avaliação geral;

- c. **administrador** – administra o ambiente, criando condições propícias para seu bom funcionamento. Tarefas típicas: cadastramento de informações, validação de matrículas, criação de autorizações de acesso (*login* e senha), verificação do funcionamento e desempenho dos recursos de hardware e software instalados;
- d. **aluno** - pode acessar os recursos do sistema : material instrucional, Fórum, Chat, Atividades, Perfil, Mediateca e Mensagens. O aluno pode também criar a sua base hipermídia, atuando como autor da mesma e podendo autorizar o acesso a outros alunos;
- como ferramenta de colaboração o sistema disponibiliza aos alunos o mesmo mecanismo de construção da estrutura hipermídia colocada à disposição dos professores. Desta forma alunos, ou grupos de alunos poderão montar a sua base de conhecimento, com acesso controlado, e desenvolver trabalhos conjuntos, onde todos os elementos do grupo poderão participar da montagem da estrutura de informações hipermídia. Uma aplicação típica dessa ferramenta é o desenvolvimento de projetos em equipes.
- O modelo do domínio foi projetado para suportar a inclusão de elementos de relacionamento semântico entre conceitos bem como a inclusão condicional de fragmentos (páginas) na página final de saída, preparando o sistema para receber futuramente o módulo adaptativo.

Informações adicionais sobre o sistema protótipo Uniclick2000 podem ser obtidas no Apêndice A.

#### 7.9.4 Considerações gerais sobre o sistema protótipo.

O sistema protótipo construído concluiu a primeira parte do sistema Uniclick2000. Foi possível verificar que a estrutura interna do modelo do domínio, mapeada para autômatos com saída, foi bastante vantajosa em termos de facilidades de implementação além das vantagens já citadas no corpo da Tese.

Uma das características importantes, que ficou evidenciada em termos de utilização

nesta etapa, foi a possibilidade de se criar o material instrucional hipermídia de forma independente do autômato. Como consequência, o material instrucional é construído em módulos e armazenado num banco de páginas central a partir do qual pode ser recuperado para a construção de um determinado curso, sem a preocupação de alteração de links. Essa facilidade, colocada para o professor autor e também disponibilizada para os alunos como ferramenta de colaboração, é de fundamental importância num ambiente de aprendizagem. Outra característica importante, também evidenciada na utilização, foi a possibilidade de se criar, para um mesmo curso, diferentes sequências de material instrucional (um curso pode ter um ou mais autômatos associados).

A arquitetura computacional em três camadas possibilitou facilidades na criação e manutenção dos módulos. O programação orientada a objetos permitirá que na fase seguinte sejam incorporados os mecanismos de adaptação, previstos na arquitetura proposta neste trabalho.

## 8 CONCLUSÕES FINAIS

Ambientes de aprendizagem na Web assumem uma importância cada vez maior no contexto do mundo atual globalizado, porque podem viabilizar a meta de levar a educação para um grande número de pessoas, rapidamente, independente de tempo e lugar, com baixo custo. O grande desafio é fazer que esses ambientes possam interagir com os usuários, estudantes e professores, com mais inteligência. A hipermídia adaptativa é uma nova área de pesquisa que procura responder a esse desafio, proporcionando novas possibilidades no processo de ensino e aprendizagem.

Cumprindo os objetivos deste trabalho, foi apresentada uma abordagem, unificando o uso de sociedade de agentes e autômatos com saída em Sistemas de Hipermídia Adaptativa (SHA), tendo sido proposta uma arquitetura geral para o esquema de adaptação conforme os requisitos definidos.

Para suportar os SHA, foi proposta uma arquitetura computacional do tipo cliente servidor, em três camadas, para suportar os SHA, com facilidades para a implementação de modelos baseados em agentes.

No escopo do trabalho foi desenvolvido um ambiente de aprendizagem hipermídia, como protótipo, utilizando o conceito de autômatos finitos com saída na estruturação do modelo do domínio. Este sistema protótipo serviu para testar e validar ambientes e ferramentas disponibilizadas na Web bem como metodologias relacionadas com a hipermídia.

Com relação à arquitetura proposta para SHA (capítulo sete), faz-se as seguintes considerações:

1. **Agentes** – a junção de *tecnologias da Web* com *tecnologias baseadas em agentes* formam uma nova tendência na modelagem e desenvolvimento de ambientes de aprendizagem.

A arquitetura proposta para o esquema de adaptação, aplica o conceito de sociedade de agentes para obter, principalmente, ganhos em flexibilidade, modularidade e expansibilidade e interoperabilidade. Artigos mais recentes tem referenciado a utilização do paradigma de agentes para a construção de SHA e STI, como uma solução promissora, principalmente em sistemas de natureza distribuída que requerem a construção de módulos autônomos e de fácil adaptação nos ambientes.

A arquitetura proposta é aberta e fornece facilidades para a inclusão de agentes para executar funções auxiliares, como por exemplo, edição e validação de regras de adaptação, funções específicas para a modelagem do usuário incluindo o gerenciamento de esterótipos, execução e avaliação de testes de avaliação vinculados ao domínio etc.

A **utilização de agentes de interface** permite a intensificação de aspectos pedagógicos no ambiente, o que favorece um melhor atendimento aos usuários. Os agentes de interface podem trabalhar de forma proativa e colaborativa com os usuários alunos executando tarefas de apoio (agentes dirigidos por objetivos) ou então tarefas auxiliares procurando informações, agendando encontros etc. No caso, a idéia de se usar agentes de interface combina perfeitamente com os objetivos de um SHA. A arquitetura proposta admite, pela sua modularidade, a inclusão de outros agentes para agirem como auxiliares no processo de interface. Este fator é muito importante para os SHA que demandam um grande esforço no seu desenvolvimento.

2. **Autômatos e a estruturação do domínio** - a estruturação do domínio do conhecimento utilizando mapeamento para autômatos é fator de fundamental importância na arquitetura proposta. O sistema protótipo, desenvolvido no escopo do trabalho, permitiu avaliar e validar a utilidade da estrutura de autômatos no contexto de um sistema de hipermídia (capítulo sete e apêndice um). Cabem aqui as seguintes considerações, relacionadas com esquemas adaptativos em SHA:

- As primitivas de modelagem do domínio, envolvendo conceitos e relacionamentos, são suportadas pelo autômato com saída.
- As funções de saída, responsáveis pela finalização das páginas Web a serem visualizadas pelo usuário, podem ser construídas (extensões) para controlar a inclusão condicional de fragmentos (adaptação de conteúdo), aspectos de layout (folhas de estilo) e ainda controlar anotação, ativação/desativação de links, sob o comando do agente de adaptação que compõe a arquitetura geral.

- Diferentes seqüências de material instrucional, construídas pelo professor autor, para um determinado curso (um autômato para cada seqüência), podem ser controladas pelo agente de adaptação em tempo de execução, em conformidade com o perfil do aluno. Ou seja, o sistema poderá recomendar ao aluno uma determinada versão do curso de acordo com o seu nível de conhecimento. Essa é uma característica importante que pode ser considerada como uma opção a mais no processo de adaptação em SHA.
- Finalmente, novas perspectivas para a utilização de autômatos poderão ser exploradas no contexto da Teoria das Categorias, incluindo operações sobre autômatos, nós e links. Esse assunto não foi tratado no escopo deste trabalho.

## 8.1 Contribuições do trabalho

Sistemas de hipermídia adaptativa constitui uma área de pesquisa recente. A maioria dos trabalhos mais importantes propõe soluções ora valorizando aspectos de modelagem, ora aspectos de navegação e preparação do material hipermídia, ora aspectos relacionados com o uso de técnicas de inteligência artificial. Como foi visto nos capítulos anteriores, o uso de autômatos com saída, por exemplo, é proposto por alguns autores para a modelagem de sistemas de hipertexto/hipermídia estáticos, sem referências ou sem aprofundamento maior sobre aspectos relacionados com a questão da hipermídia adaptativa. Por outro lado, especificamente para o processo de adaptação em SHA, em geral, não há referências explícitas sobre o uso da tecnologia de agentes unificada com autômatos.

Nesse contexto, a principal contribuição deste trabalho está exatamente em evidenciar as potencialidades de autômatos e agentes em resolver juntos problemas da hipermídia adaptativa, culminando com a proposição de uma arquitetura geral para SHA.

A segunda contribuição está relacionada com o sistema protótipo. Nesse sistema o modelo do domínio é mapeado para autômatos e os resultados apontam para novas perspectivas para ambientes de aprendizagem, principalmente em relação às facilidades oferecidas para o professor na autoria dos cursos, bem como para os alunos no uso de recursos de colaboração.

Outra contribuição que julgamos importante é de caráter investigativo, envolvendo a pesquisa bibliográfica sobre o assunto.

## **8.2 Trabalhos futuros**

A partir dos estudos e proposições constantes desta Tese, vislumbramos os seguintes trabalhos de continuidade:

- Desenvolvimento dos módulos de adaptação do sistema Uniclick2000, utilizando diversos métodos e técnicas de SHA. Este trabalho pela sua complexidade e extensão, pode ser dividido em partes tratando do modelo do usuário, a navegação adaptativa, a adaptação de conteúdo, o editor de regras pedagógicas e a máquina de inferência.
- No caso específico do modelo do usuário cabem estudos complementares para avaliar diferentes técnicas para a modelagem.
- Realizar estudos sobre a Teoria das Categorias e sua aplicação em SHA.

## REFERÊNCIAS

ANJANEYULU, K. S. R. **Concept level modelling on the WWW**. 1997. Disponível em: <[http://www.contrib.andrew.cmu.edu/~plb/AIED97\\_workshop/anjaneyulu.html](http://www.contrib.andrew.cmu.edu/~plb/AIED97_workshop/anjaneyulu.html)>. Acesso em: out.2000.

ALMEIDA, Maria A. F. **Aprender, atividade inteligente: e se esta inteligência for parcialmente artificial?** Florianópolis, 1999. 102 p. Dissertação (Mestrado em Ciência da Computação). Programa de Pós-Graduação em Ciência da Computação, Universidade Federal de Santa Catarina.

ARMSTRONG, R.; FREITAG, D. **WebWatcher: A learning Apprentice Of the World Wide Web**, 1995. Disponível em: <<http://www.cs.cmu.edu/afs/cs.cmu.edu/project/theo-6/web-agent/www/webagent-plus/webagent-plus.html>> . Acesso em: out. 2000.

BALASUBRAMANIAN, V. **State of the Art Review on Hypermedia Issues and Applications**, 1994. Disponível em: <<http://netspot.city.unisa.edu.au/netspot/tutorial/hypertext.html>; <http://netspot.city.unisa.edu.au/netspot/tutorial/hypertext.html>>. Acesso em: dez. 2001.

BARRETO, J. M. **Notas de aulas da disciplina Modelagem e Simulação de Sistemas Físicos e Biológicos**, Mestrado em Engenharia Elétrica, Florianópolis, GPB-UFSC, 1995.

BEAUMONT, J.H. User Modeling in the Interactive tutoring system ANATOM-TUTOR. **User Modeling and User-Adapted Interaction**, Dordrecht: Kluwer Academic Publishers, 1994. p. 21-45.

BECK, J. E.; STERN, M. K.; HAUGSJAA, E. Applications of AI in Education. **ACM Crossroads**, Set. 1998. Disponível em: <http://www.acm.org/crossroads/xrds3-1/aied.html>. Acesso em: set. 2000.

BOYLE, C.; ENCARNACION, A. O. MetaDoc : An adaptive hypertext Reading System. **Journal of user modeling and user adapted interaction**, Dordrecht, v.4, n.1, p. 1-19, 1994.

BRADSHAW, Jeffrey M. **Software Agents**. Menlo Park. CA: The MIT Press, 1997.

BREITMAN, K. **Hiper-Autor: Um Método para Especificação e Projeto de Sistemas Hypermedia**. Rio de Janeiro, 1993. 120 p. Dissertação (Mestrado em Engenharia de Sistemas) – Programa da Coppe-Sistemas, UFRJ.

BRENNER, VALTER **Intelligent Softwares Agents**. Berlim: Springer Verlac, 1998.



BRUSILOVSKY, P. A study of user model based link annotation in educational hypermedia. **Journal of Universal Computer Science**, Berlin, abr.1998.v. 4, n. 4, p. 428 – 448.

BRUSILOVSKY, P. **Adaptive and Intelligent Technologies for Web-based Education**, 1999. Disponível

em:<<http://www.contrib.andrew.cmu.edu/~plb/papers/KI-review.html>>.

Acesso em: dez. 1999.

BRUSILOVSKY, P. Efficient techniques for adaptative hypermedia.

In: NICHOLAS, C.; MAYFIELD, J. (Eds). **Intelligent Hypertext: advance techniques for WWW**. Berlin: Springer Verlac, 1997. p. 12-30.

BRUSILOVSKY, P. Intelligent Tutor, Environment and Manual for Introductory Programming. **Educational and Training Technology International**, Berlim, 1992. p.26-34.

BRUSILOVSKY, P. Methods and techniques of adaptive hypermedia. **User Modeling and User Adapted Interaction**, Dodrecht, v. 6, n. 2-3, p. 87-129, 1996.

BRUSILOVSKY, P.; PESIN, L. **ISIS-Tutor: An Adaptive Hypertext Learning Environment for CDS/ISIS users**, 1994. Disponível em:

<[http://cs.joensuu.fi/~mtuki/www\\_clce.270296/Brusilov.html](http://cs.joensuu.fi/~mtuki/www_clce.270296/Brusilov.html)>. Acesso em: out.2000.

BRUSILOVSKY, P.; SCHWARZ, E. User as student: Towards an adaptive interface for advanced web-based applications. **Proceedings of the sixth international conference on user modeling**, UM97, Sardinia, Italy, 1997.

BRUSILOVSKY, P.; SCHWARZ, E.; WEBER, G. A tool for developing adaptive eletronic textbooks on WWW. **Proceedings of webnet/96. world conference of the web society**. Boston, MA, 1996a.

BRUSILOVSKY, P.; SCHWARZ, E.; WEBER, G. **ELM-ART: An Intelligent Tutoring System on World Wide Web**. In: CLAUDE FRASSON, GILLES GAUTHIER, AND ALAN LESGOLD, EDITORS, **Intelligent Tutoring Systems, v.1086 of Lecture Notes in Computer Science**, 1996. p. 261-269.

BRUSILOVSKY, Peter; EKLUND, John; SCHARWZ, E. Web based education for all: a tool for development adaptive courseware. In: INTERNATIONAL WORLD WIDE WEB CONFERENCE, 7., 1998, Brisbane. **Proceedings...** Brisbane: Elsevier Science, 1998.

Disponível em <<http://www7.scu.edu.au/>>.

Acesso em: set. 2000.

BUSH, V. **As We May Think**. The Atlantic Monthly, Jul. 1945. Disponível em:

<<http://www.isg.sfu.ca/%7Educhier/misc/vbush>; <http://www.isg.sfu.ca/~duchier/misc/vbush>>. Acesso em: mar. 1999.

CAGLAYAN, A; HARRISON, C. **Agent Sourcebook**. Wiley Computer Publishing,

John Wiley, 1997.

CAMPBELL, Brad; GOODMAN, Joseph. HAM: A general purpose hypertext abstract machine. **Communications of the ACM**, New York, v.31, n.7, p.856-861, Jul. 1998.

CHAIBEN, Hamilton. **Hipermídia na educação**, 2001. Disponível em: <[http://www.cce.ufpr.br/~Hamilton/hed\\_00001.htm](http://www.cce.ufpr.br/~Hamilton/hed_00001.htm)>. Acesso em: out. 2001.

CARRO, R.M; PULIDO, E; RODRÍGUEZ, P. **TANGOW: a Model for Internet Based Learning**. Disponível em: <<http://www.inderscience.com/ejournal/c/ijceell/ijceell2001/ijceell2001v11n12.html>>. Acesso em: nov. 2001.

CONATI, Cristina; VANLEHN, Kurt. **Pola: a student modeling framework**. Disponível em: <http://www.cs.ubc.ca/~conati/my>. Acesso em: dez. 2000.

COSTA, M. T. C. **Uma Arquitetura Baseada em Agentes para Suporte ao Ensino Distância**. Florianópolis, 1999. 97 p. Tese (Doutorado em Engenharia de Produção) – Programa de Pós-Graduação em Engenharia de Produção, Universidade Federal de Santa Catarina.

DE BRA, P.; AERTS, A.; HOUBEN, GJ; WU, H. **Making General-Purpose Adaptive Hypermedia Work**. Eindhoven University of Technology, 1998. Disponível em : <[http://wwwis.win.tue.nl/~alex/paper\\_collection/webnet00.pdf](http://wwwis.win.tue.nl/~alex/paper_collection/webnet00.pdf)>. Acesso em: dez.2000.

DE BRA, P.; CALVI, L. **AHA: a Generic Adaptive Hypermedia System**. Eindhoven University of Technology, 1999. Disponível em: <<http://wwwis.win.tue.nl/ah98/DeBra.html>>. Acesso em: dez. 2000.

DE BRA, Paul; HOUBEN, G.J; KORNATZKY, Y. **An Extensible Data Model for Hyperdocuments**, 1992. Disponível em: <<http://citeseer.nj.nec.com/cache/papers/cs/1919/http:zSzzSzwwwis.win.tue.nlzSz~debrazSzecht92zSzfinal.pdf/debra92extensible.pdf>> Acesso em: dez. 1999.

DE BRA, Paul. **Design Issues in Adaptive Web-Site Development**. Paul De Bra Eindhoven University of Technology The Netherlands, 1999. Disponível em: <<http://wwwis.win.tue.nl/~debra/asum99/debra/debra.html>>. Acesso em: jan. 2000.

DE BRA, Paul; BRUSILOVSKY, P. ; HOUBEN, G. Adaptive Hypermedia: from Systems to Framework. **ACM Computing Surveys**, dez. 1999. Disponível em: <<http://www.acm.org/surveys/Formatting.html>>. Acesso em: dez. 2001.

DE BRA, Paul; CALVI, Licia. Creating adaptative hyperdocuments for and on the web. In: **WEBNET – WORLD CONFERENCE OF THE WWW, INTERNET AND INTRANET**. Charlottesville, 1997. **Proceedings...** Charlottesville: Association for the Advancement of Computing Education, 1997.

DE BRA, Paul; HOUBEN, G.; WU, H. **AHAM: a Dexter-based reference model for adaptive hypermedia**. UK CONFERENCE IN HYPERTEXT, 1999. Disponível em: <<http://citeseer.nj.nec.com/debra99aham.html>>. Acesso em: dez. 2000.

DE ROSIS, F.; DE CAROLIS, N.; PIZZUTILO, S. **User Tailored Hypermedia Explanations**. Disponível em : <<http://www.cs.bgsu.edu/hypertext/addaptive/deRosis.html>>

Acesso em: dez. 2000.

ENGELBART, D. C. A Conceptual Framework for the Augmentation of Man's Intellect. In: HOWERTON, P. D.; WEEKS, D. C. (Ed.), *Vistas in Information Handling*. v. 1, Spartan Books, p. 1-29, 1963.

FARACO, Rafael Ávila. **Uma arquitetura de agentes para negociação dentro do domínio do comércio eletrônico**. Florianópolis, 1998, 87 p. Dissertação (Mestrado em Engenharia de Produção) – Programa de Pós-Graduação em Engenharia de Produção, Universidade Federal de Santa Catarina.

FISCHER, G et al. Minimalist explanations in knowledge-based systems. In: ANNUAL HAWAII INTERNATIONAL CONFERENCE ON SYSTEM SCIENCES, 23. , 1990, Kailua-Kona. **Proceedings...** Los Alamitos: IEEE Computer Society Press. p. 309-307.

FRANKLIN S.; GRAESSER, A. Is it a Agent, or just a Program? A Taxonomy for Autonomous Agents. **Proceedings of the 3<sup>rd</sup> international workshop on agent theories**, Springer-Verlag, 1996.

FUJIHARA, H.; SNELL, J.; BOYLE, C.D. Intelligent Search in an Educational Hypertext Environment. **Journal of Educational Multimedia and Hypermedia**, 1(4), p. 401-415, 1992.

GARZOTTO, F.; MAINETTI, L. ; PAOLINI, P. Hypermedia Design Model 2. **Communication of the ACM**, v. 38, n. 8, p. 74-8, 1995.

GARZOTTO, F.; PAOLINI, P.; SCHWABE, D. HDM: a Model-Based Approach to Hypermedia Application Design. **ACM Transactions Information system**, v. 11. n.1, p. 1-26, 1993.

GENESERETH M. R.; KETCHPEL, S. P. Software Agents. **Communications of the ACM**, Jul. 1994, p. 48-53, 147.

GIRAFFA, L.M.M. **Uma arquitetura de tutor utilizando estados mentais**, Porto Alegre, 1999, 166 p. Tese (Doutorado em Ciência da Computação) - Universidade Federal do Rio Grande do Sul, UFRGS.

GONSCHOREK, M; HERZOG, G. Using hypertext for an adaptive helpsystem in an intelligent tutoring system. In: AI-ED WORLD CONFERENCE ON ARTIFICIAL INTELLIGENCE IN EDUCATION, 7, 1995, Washington. **Proceedings ...** Charlottesville: AACE, 1995. p. 274-281.

GRUNST, G. Adaptive hypermedia for support systems. In: SCHNEIDER H., KUHME,

T; MALINOWSKY, U (EDS). **Adaptive user interfaces: principles and practice**. Amsterdam: North-Holland, 1993. p. 269-283.

HALASZ, Franz; SCHWARTZ, Mayer. The Dexter hypertext reference model. In: **HYPERTEXT STANDARDIZATION WORKSHOP**, 1990, Gaithersburg. **Proceedings...** Gaithersburg: NIST Special Publication, 1990. p. 150-178.

HENZE, NICOLAS. **Adaptive Hyperbooks: Adaptation for Project Learning Based Resources**. Hannover, 2000. 95 p. Tese (Doutorado). Universidade de Hannover. Disponível em: <<http://edok01.tib.uni-hannover.de/edoks/e002/313646791.pdf>>. Acesso em: Nov. 2001.

HOHL, H.; BÖCKER, H.; GUNZENHÄUSER, R. Hypadapter: An adaptive hypertext system for exploratory learning and programming. **User, Modeling and User-Adapted Interaction**, Dordrecht, v. 6, p. 131-156, 1996.

HOOK, K et al. A glassbox approach to adaptive hypermedia. In: **Brusilovisky, Peter et al. adaptive hypertext and hypermedia**. Dordrecht: Kluwer Academic Publishers, 1998. p. 143-170.

HOPCROFT, J.E.; ULMANN, J. D. **Introduction to automata theory, languages and computation**. Addison-Wesley, 1979.

ISAKOWITZ, T.; STOHR, E; BALASUBRAMANIAN, P. Relationship Management Methodology. **Communication of the ACM**, v.. 38, n.8, p. 34-48, 1995.

JONASEN, D.H. Designing Structured Hypertext and Structuring Access to Hypertext. **Educational Technology**, p.13-16, Novembro, 1988.

KAPLAN, C.; FENWICK, J.; CHEN, J. Adaptive Hypertext Navigation Based on User Goals and Context. **User Models and User Adapted Interaction**, 3(3), p. 193-220, 1993.

KAY, J.; KUMMERFELD, R. **Adaptive hypertext for individualised instruction**. Disponível em: <http://www.cs.bgsu.edu/hypertext/adaptive/Kay.html>. Acesso em: dez. 1999.

KAY, J.; KUMMERFELD, R. User Models for Customized Hypertext. In: **intelligent hypertext-advanced techniques for the world wide web**, c. Nicholas and J. Mayeld, Ed., LNCS V.1326. Springer, 1997.

KOBSA, A.; MÜLLER, D.; NILL, A. KN-AHS: an adaptive hypertext client of the user modeling system BGP-MS. In: **FOURTH INTERNATIONAL CONFERENCE ON USER MODELING**, HYANNIS, Massachusetts, USA, August 1994, Hyannis. **Proceedings...**Hyannis: Mitre Corporation, 1994. p. 31-36.

LANGE, Danny B. A formal model of hypertext. In: MOLINE, Judi et al. In: **HYPERTEXT STANDARDIZATION WORKSHOP**, GAITHERSBURG. **Proceedings...**Gaithersburg: NIST Special Publication, v. 500-178, 1990. MD 20899, 1990.

LAROUSSI, M. **Conception et Réalisation d'un Système Didactique Hypermédia Adaptatif: Camaleon**, 2001. Tese (Doutorado) – Université Manouba. Disponível em: <http://www.wis.win.tue.nl/ah/thesis.html>  
Acesso em: dez. 2001.

MACHADO, J. H. A. P. **Hyper-Automaton: hipertextos e cursos na Web usando autômatos finitos com saída**. Porto Alegre, 2000, 137 p. Dissertação ( Mestrado). Mestrado em Ciência da Computação. Programa de Pós-Graduação em Computação da Universidade Federal do Rio Grande do Sul.

MACHADO, J. P et al. **Autômatos finitos: um Formalismo para Cursos na Web**. 1999. Disponível em: [www.inf.ufsc.br/sbes99/anais/SBES-Completo/20.pdf](http://www.inf.ufsc.br/sbes99/anais/SBES-Completo/20.pdf) . Acesso em: dez. 2000.

MAES, P. The Agent Network Architecture. **Proceedings of th 1991 AAAI spring symposium on integrated intelligent architectures**, AAAI Press, Stanford, 1991.

MAES, P. Agents that Reduce Work and Information Overload. **Communications of the ACM**, v.37, n.7, p. 31-40, 146, ACM Press, July 1994.

MAES, P. **Modeling Adaptive Autonomous Agents**. MIT Media-Laboratory, CA., 1995.

MAIDANTCHIK, C; XEXEO, G. B. **Hyperdev: hypertext tool to support object oriented software development**. Coppe, Universidade Federal do Rio de Janeiro, 1996. Disponível em: <http://www.cos.ufrj.br/~lodi/HyperDev/hyper.html>. Acesso em: out. 2001.

VANLEHN, Kurt . **Olae: A Bayesian performance assessment for complex problem solving**, 1998. Disponível em: <http://www.pitt.edu/~vanlehn/distrib/NCMEOlae3.html>. Acesso em: dez. 2000.

MATHÉ, N; CHEN, J. User –centered indexing for adaptive information access. **User modelling and user-adapted interaction, special issue on adaptive hypertext and hypermedia**, Dordrecht, v. 6, n. 2-3, 1996.

MAURER, Hermann. Necessary ingredients of integrated network based learning environments. In: ED-MEDIA/ED-TELECOM EDUCATIONAL MULTIMEDIA/HYPERMEDIA AND TELECOMMUNICATIONS, 1997, Calgary. **Proceedings...** Charlottesview: AACE, 1997. v. 1, p. 619-624.

MICARELLI, A.; SCIARRONE, F. A case-based toolbox for guided hypermedia navigation. In: CONFERENCE ON USER MODELING, UM'96, 5., 1996, Kailua, Hawaii. **Proceedings...** Kailua Kona: User modeling Inc., 1996. p.129-136.

MOREAU, Luc. Hall, Wendy. **On the Expressiveness of Links in Hypertext Systems**. 1998. Disponível em: <http://eprints.ecs.soton.ac.uk/archive/00002753/>. Acesso em: dez. 1999.

MUKHERJEA, S.; FOLEY, J. D. Visualising the WWW with a navigational view build. Disponível em: <http://www.igd.fhg.de/www/www95/proceedings/papers/44/mukh/mukh.html>. Acesso em: dez. 1999.

NANARD, J.; NANARD, M. Hypertext Design Environment and Hypertext Design Process. **Communication of the ACM**, v. 38, n.8, p. 49-56, 1995.

NELSON, T. **Literary Machines**. Sausalito, Califórnia: Mindful Press, 1993.

NORVIG, S.; RUSSEL, S. **Artificial Intelligence: a modern approach**. New Jersey: Prentice Hall, 1995.

NWANA, H. Software Agents: An Overview. **Knowledge Engineering Review**, v. 11, n. 3, p. 1-40, set. 1996.

PEREIRA, M. **Um modelo distribuído com arquitetura 3 camadas baseado em agentes para suporte ao ensino**. Florianópolis, 1999, 98 p. Dissertação (Mestrado em Engenharia de Produção) – Programa de Pós-Graduação em Engenharia de Produção, Universidade Federal de Santa Catarina.

PEREZ, T.; GUTIERREZ, J.; LOPISTÉGUY, P. An adaptive hypermedia system. In: AI-EWD World CONFERENCE ON ARTIFICIAL INTELLIGENCE IN EDUCATION, 7, 1995, Whashington. **Proceedings....** Charlottesville: AACE, 1995. p.351-358.

ROSATELLI, M. C. Novas Tendências na Pesquisa em Inteligência Artificial na Educação. **Escola de Informática da SBC**. Ijuí, RS, 2000.

ROSIS, F. D. et al. Modeling the User Knowledge by Belief Networks. *User Modeling and User Adapted Interaction*, 1992. v. 2, p. 367-388.

SANTOS, N. **Navegação em Documentos Hipermídia: Estado de Arte, 1996**. Disponível em: <<http://www.ime.uerj.br/professores/neide/index.htm>>. Acesso em: nov. 2001.

SCHWABE, D.; ROSSI, G. Object-Oriented Hypermedia Design Method. **Communication of the ACM**, v. 38, n.8, 45-46, 1995.

SHOHAM, Y. **A Overview of Agent-Oriented Program**. In: SOFTWARE AGENTS, ED. J. M. BRADSHAW, California: AAI Press, 1997.

SHOHAM, Y. Agent oriented programming. **Artificial Intelligence**, v. 60, n.1, p. 51-92, 1993.

SICHMAN, J.; DEMAZEU, Y.; BOISSIER, O. How can knowledge-based systems be called agents? In: SIMPÓSIO BRASILEIRO DE INTELIGÊNCIA ARTIFICIAL, 9., 1999, Rio de Janeiro. **Proceedings...** Rio de Janeiro: Pontifícia Universidade Católica do Rio de Janeiro, 2000. p. 173-185.

SILVA, D. P.; DURM, R.V.; DUVAL, E.; OLIVIE, H. **Concepts and documents for adaptive educational hypermedia: a model and a prototype**. 1998. Disponível em: <http://www.wis.win.tue.nl/ah98/Pilar/Pilar.html>>. Acesso em: dez. 2000.

STOTTS, P. David; FURUTA, Richard. Petri-Net based hypertext: Document structure with browsing semantics. **ACM Transactions on Information Systems**, New York, v.7, n.1, p. 3-29, 1989.

STOTTS, P. David, FURUTA, R.; CABARRUZ, Cyrano Ruiz : Hyperdocuments as Automata: Verification of Trace-Based Browsing Properties by Model Checking. **ACM Transactions on Information Systems**, v. 16, n. 1, p. 1-30, 1998.

THIRY, M. **Uma Arquitetura Baseada em Agentes para Suporte ao Ensino à Distância**. Florianópolis, 1999, 97 p. Tese (Doutorado em Engenharia de Produção) – Programa de Pós-Graduação em Engenharia de Produção, UFSC.

TOMPA, F. W. M. A data model for flexible hypertext database systems. **ACM transactions on information systems**, n. 7, p. 85-100, 1989.

THURING, M; HANNEMANN J.; HAAK, J. Hypermedia and cognition: designing for comprehension. **Communications of the ACM**, 38(8), p. 57-66, 1995.

VASSILEVA, J. A task-centered approach for user modelling in a hypermedia office documentation system. **User Modeling and User-Adapted Interaction**. Special issue on adaptive hypertext and hypermedia, Dordrecht, v.6, n.2-3, 1996. p. 87-129.

VICCARI, R.M. Automatic Symbolic Learning. **RITA - Revista de Informática Teórica e Aplicada**, UFRGS, n. 1, 1990.

WATERWORTH, J. A. A pattern of islands: exploring public information space in a private vehicle. In: BRUSILOVSKY, P.; KOMMERS, P.; STREITZ, N. (Eds.). **Multimedia, Hypermedia and Virtual Reality**. Berlin: Springer-Verlag, 1996.

WEBER, G.; SPECHT, M. User modeling and adaptive navigation support in WWW-based tutoring systems. **Proceedings of the sixth international conference on user modeling, UM97**, Sardinia, Italy, 1997.

WEISS, Gerhard. **Multiagents systems: a modern approach to distributed artificial intelligence**. Mit Press, 1999.

WHALLEY, P. An Alternative Rhetoric for Hypertext. In: C. MCKNIGHT; A. DILLON; J. RICHARDSON. **Hypertext, a psychological perspective**. Ellis Horwood, 1993.

WOOLDRIDGE, M.; JENNINGS, N. R. Intelligent Agents: Theory and Practice. **Submitted to the Knowledge Engineering Review**, v.10, 1995.

WU, Hongjing; HOUBEN, Geert-Jan, DE BRA, Paul de. **AHAM: A Reference Model to Support Adaptive Hypermedia Authoring**, 1999. Disponível em: <http://www.wis.win.tue.nl/~debra/asum99/debra/debra.html>>. Acesso em: dez. 2001.

ZHONGMIN, Li; O'NEIL, Harold F.; BAKER, Eva L. Developing a Research Reference Interface for Knowledge-Based Instructional Design Tools. **Educational Technology**, August 1991.



## **APÊNDICES**

## **APÊNDICE A - AMBIENTE DE APRENDIZAGEM BASEADO EM HIPERMÍDIA ADAPTATIVA - PROTÓTIPO**

### **1 Introdução**

Este documento descreve as principais características do ambiente de aprendizagem hipermídia denominado Uniclick2000 (protótipo), desenvolvido no escopo desta Tese, no Laboratório de Projetos do Curso de Ciência da Computação da Unisul – Campus de Tubarão.

O desenvolvimento do protótipo foi dividido em duas partes; na primeira parte o desenvolvimento e implementação de um sistema de hipermídia; na segunda parte, desenvolvimento e implementação dos módulos de adaptação (modelagem do usuário e módulo de adaptação), completando os requisitos do projeto. No contexto desta Tese foi possível concluir a primeira parte do projeto. No entanto a estruturação do software, principalmente na questão do modelo do domínio do conhecimento/ estrutura de navegação (utilizando autômatos) e arquitetura computacional, foi preparada para a inclusão dos módulos de adaptação.

### **2 Rápido histórico**

A primeira versão do sistema foi desenvolvida em 1999. Nessa versão, o foco foi apoiar, via Internet, os cursos presenciais de graduação de uma Universidade. Essa versão foi utilizada em caráter experimental em quatro disciplinas do curso de Ciência da Computação no período de 2000/2001, envolvendo cerca de 80 alunos ativos em cada semestre, tendo sido obtida boa aceitação por parte dos alunos. Na primeira versão, o sistema apresenta as seguintes características funcionais: arquitetura computacional em três camadas, com

armazenamento dinâmico de todas as informações em banco de dados relacional; funcionamento multicampi; controle de acesso diferenciado para usuários alunos, professores e administradores; o aluno tem acesso apenas nas disciplinas nas quais está matriculado; professores e alunos possuem áreas distintas de publicação; a operação de publicação é facilitada, podendo ser publicados os principais tipos de arquivos sem a necessidade de transformação em arquivo HTML; ferramentas de chat e fórum são disponibilizadas para alunos e professores. Em resumo, o sistema permite ao professor disponibilizar de maneira simples o material instrucional via *Web*.

O aprendizado, com a primeira versão foi um passo importante no contexto da Tese, não somente em relação aos aspectos tecnológicos envolvidos como também em relação às questões de ordem pedagógica.

A segunda versão do sistema foi desenvolvida no período 2000/2001, originando o sistema Uniclick2000, com uma reformulação total em termos de projeto: o novo objetivo foi o desenvolvimento de um ambiente de aprendizagem baseado em hipermídia adaptativa. Esta versão é objeto dos itens a seguir neste documento.

### **3 Ferramentas utilizadas**

Para efeito de estudos e de melhor avaliação das tecnologias, no decorrer do projeto foram utilizadas as ferramentas:

- Ambiente ASP - Active Server Pages (ASP), com o servidor Web IIS 4.0.
- Ambiente Java: JDK 1.1.8, JSDK 2.0, Apache Web Server 1.3.6 com Jserv.
- Banco de dados Oracle 8.0.
- Sistemas operacionais nos servidores: Windows NT 4.0, Red Hatt Linux 6.0.

Muito embora o ambiente ASP tenha apresentado, pela sua simplicidade, facilidades para o aprendizado e utilização, prevaleceu o ambiente JAVA em função das potencialidades relativas à portabilidade, processamento distribuído e características da própria linguagem de programação JAVA na metodologia de orientação a objetos.

### **4 Módulos do Sistema Uniclick 2000 (protótipo da Tese)**

Os módulos do Sistema Uniclick 2000 estão mostrados a seguir:

<b>Gerência de informações básicas</b>	<b>Gerência de Cursos</b>	<b>Gerência de Comunicação</b>	<b>Gerência de Interface</b>
<b>Cadastro de usuários</b> Responsabilidade: usuários em geral. <b>Registro de matrícula</b> Responsabilidade: o usuário solicita a matrícula e Administrador procede a validação. <b>Cadastro de cursos</b> Responsabilidade: Administrador. <b>Cadastro de disciplinas</b> Responsabilidade: Administrador <b>Cadastro de disciplinas oferecidas</b> Responsabilidade: Administrador.	<b>Banco de páginas dos professores</b> Responsabilidade: Professor Autor <b>Banco de páginas dos alunos</b> Responsabilidade: alunos. <b>Autômato (estrutura de navegação)</b> Responsabilidade: Professor Autor <b>Definição de esquema de avaliação</b> Responsabilidade: Professor Autor <b>Avaliação de alunos</b> Responsabilidade: Professor Tutor <b>Banco de testes</b> Responsabilidade: Professor Autor <b>Banco de atividades</b> Responsabilidade: Professor Autor <b>Midioteca</b> Responsabilidade: Professor Tutor. <b>Anotações:</b> Responsabilidade: Aluno/professor.	<b>Forum</b> Responsabilidade: Gerado pelo Professor Autor e mantido pelo professor tutor e alunos. <b>Faq</b> Responsabilidade: professor autor. <b>Galeria</b> Responsabilidade: utilizado pelo aluno para publicar material didático. <b>Mural</b> Responsabilidade: professor tutor e administrador. <b>Tutoria</b> Responsabilidade: professor tutor. <b>Chat</b> Responsabilidade: usuários em geral. <b>Perfil</b> Responsabilidade: usuários em geral.	<b>Interface geral</b> log: registro de logs.

Tabela 4 – Módulos do sistema Uniclick2000

**Gerência da Interface:**

- Módulo de interface: responsável pela interface geral do sistema.
- Módulo de log: registro de log do acesso à rede hipermídia (cursos) pelos usuários.

**Gerência de informações básicas:** envolve os módulos de criação e manutenção das tabelas básicas do sistema (cadastro de usuários, de cursos e disciplinas) e validação/ registro de matrículas.

**Gerência de cursos:** é a parte central do sistema onde estão os módulos responsáveis por criar, mostrar e manter: autômatos, páginas em geral, testes com correção automática,

atividades a serem corrigidas pelo professor tutor, esquema de avaliação para o curso como um todo, mediateca, avaliação de atividades de alunos, anotações feitas por alunos e professores.

**Gerência de comunicações:** São oferecidas as seguintes ferramentas de comunicação: Fórum, Chat, Perfil, Tutoria (tira dúvidas com o professor), Galeria (publicação de trabalhos de alunos), FAC (dúvidas mais frequentes) e Mural. As ferramentas Fac e Fórum são objetos que podem ser inseridos na estrutura de nós hipermídia, a critério do professor, constituindo-se em elementos específicos ao contexto dos conceitos envolvidos na estrutura.

## 5 Principais características funcionais do Uniclick2000

Serão descritos a seguir as principais características do sistema Uniclick2000 a partir dos eventos, atores e fluxos relacionados.

### 1. Evento: Usuário deseja se cadastrar no Ambiente.

**Ator:** Usuário Geral.

**Descrição do fluxo:**

O usuário (aluno, professor autor, professor tutor, monitor), solicita cadastramento na interface do sistema. O usuário preenche seus dados (login, senha, nome, endereço, etc) e aciona a ferramenta cadastro.

Todos os usuários que se cadastram são considerados alunos inicialmente. A definição de status de professores (autores e tutores) e monitores é feita posteriormente pelo administrador.

Efetuada o cadastramento, automaticamente é efetuado logon no sistema.

A tabela abaixo indica os níveis de autorização dos usuários do sistema:

ITEM	ALUNO	TUTOR	AUTOR	ADMINISTRADOR
Cadastro de usuários	IA	IA	IA	M
Registro de matrícula	I	I	I	M
Cadastro de cursos	C	C	C	M
Cadastro de disciplinas	C	C	C	M
Cadastro de disciplinas oferecidas	C	C	C	M
Banco de páginas do professor autor	C	C	M	C
Banco de páginas do aluno	M	C	C	C
Autômatos do professor autor	C	C	M	C

ITEM	ALUNO	TUTOR	AUTOR	ADMINISTRADOR
Automatos do aluno	M	C	C	C
Definição de esquemas de avaliação	C	C	M	C
Avaliação de alunos	C	M	C	C
Banco de testes	C	C	M	C
Banco de atividades	C	C	M	C
Mediateca	C	M	C	C
Anotações	M	M	M	M
Forum	M	M	M	M
Fac	C	M	M	C
Galeria	M	C	C	C
Mural	C	M	M	M
Tutoria	M	M	M	C
Perfil	M	M	M	C

M→ Mantem (insere, altera, exclui e consulta); C→ consulta; I→ insere; A→ altera

Tabela 5 – Níveis de autorização do sistema Uniclik2000

### Exceções:

Caso o login já exista, o sistema retorna com erro.

Caso as senhas não confirmam (senha de confirmação), o sistema retorna com erro.

## UNICLICK2000

Login   
Senha   
Entrar

**Geral**  
[Cadastre-se](#)  
[Metodologia](#)  
[Noticias](#)  
[Quem Somos](#)  
[Entre em Contato](#)  
**Biblioteca**  
[Artigos](#)  
[Entrevistas](#)  
[Arquivo](#)

**Cursos Oferecidos**  
**Informática**  

- [ASP](#)
- [Curso Teste](#)

**Uniclick 2001 - Ambiente Virtual de Aprendizagem**  
**Apresentação**  

Este trabalho apresenta as principais características de um sistema que compõem um ambiente virtual de aprendizagem baseado em hipermídia. Esse sistema tem como objetivo a realização de cursos à distância, bem como, apoiar as disciplinas de cursos na modalidade presencial.

São utilizados conceitos da Ciência da Computação, em especial a teoria dos autômatos, juntamente com conceitos relativos a sistemas de informações para funcionamento na web. O ambiente permite criar cursos em hipermídia, com características de reusabilidade de páginas para outros cursos, possibilitando um maior reaproveitamento de material didático e a construção de cursos com enfoques diferenciados.

© 2001, Luiz A. Soares Garcindo

Figura 19 - Interface inicial do sistema Uniclick2000.

**2. Evento:** Aluno solicita matrícula em curso.**Ator:** Usuário Geral.**Descrição do fluxo:**

Feito cadastro do usuário no sistema é necessário agora a matrícula no curso desejado.

O usuário (aluno, professor autor, professor tutor, monitor, administrador), solicita matrícula no curso desejado. O usuário aciona a ferramenta de matrícula.

O usuário deverá esperar agora que o administrador defina o seu status (professor autor, professor autor, monitor) e valide a matrícula no curso.

Um usuário cadastrado poderá ser matricular em mais de um curso.

**Exceções:**

Caso já exista matrícula, ocorre um erro.

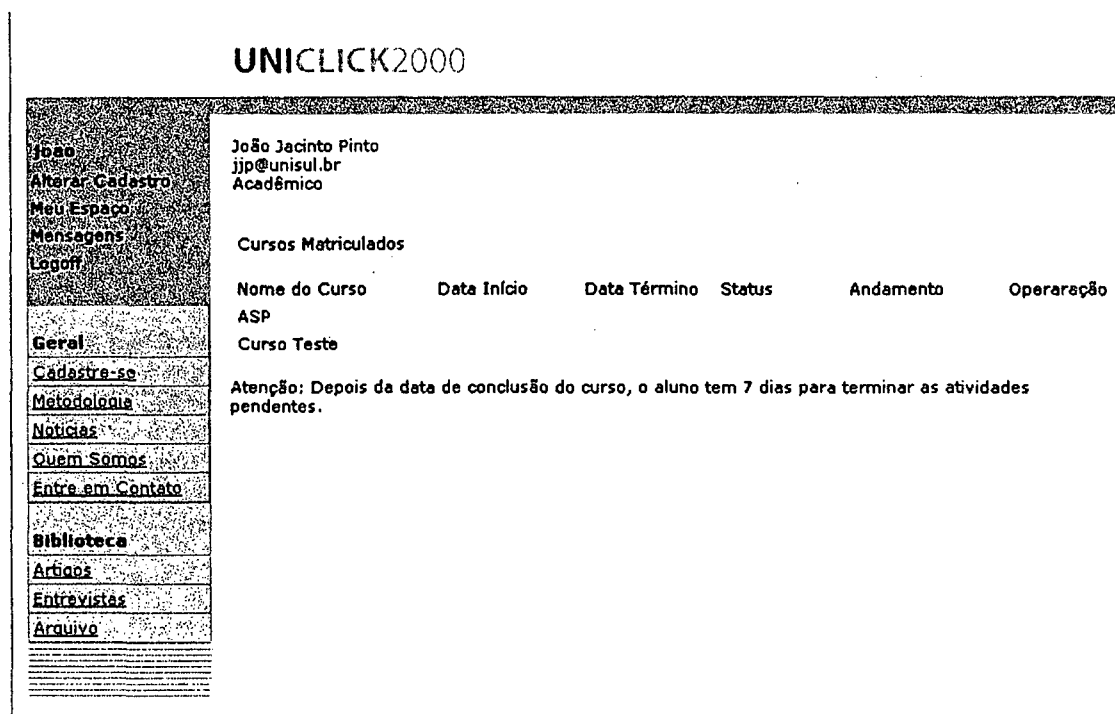


Figura 20 – Interface para visualizar matrículas em cursos

**3. Evento:** Usuário deseja acessar o ambiente de aprendizagem**Ator:** Usuário Geral

**.Descrição do fluxo:**

Para acessar o ambiente, é necessário o fornecimento de login/senha para o sistema. Feito isso, o sistema mostrará opções pertinentes ao usuário como alterar cadastro, “Espaço meu”, mensagens e logoff.

**Exceções:**

Caso Login/Senha estiverem incorretos, ocorre um erro.

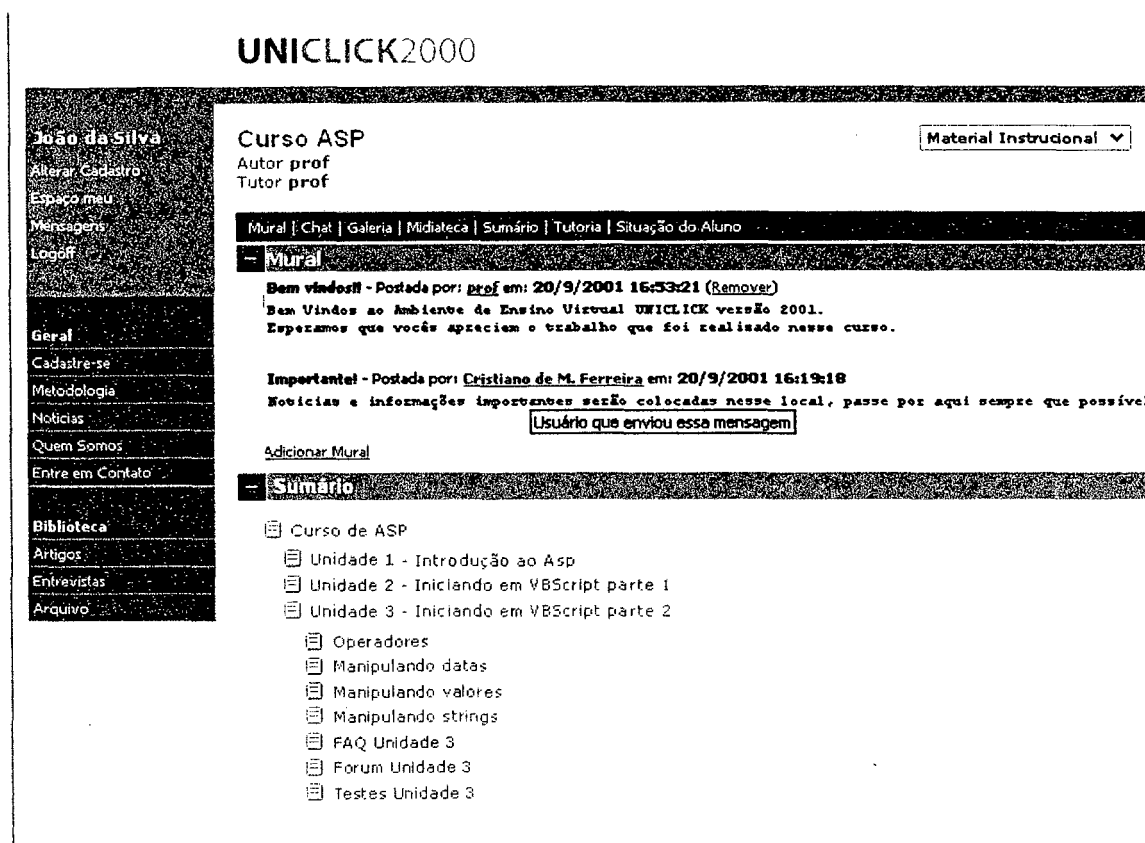


Figura 21 – Interface mostrando sumário do material instrucional para o aluno.

4. **Evento:** Usuário quer acessar seu curso.

**Ator:** Usuário Geral

**Descrição do fluxo:**

O usuário, já tendo efetuado o login no sistema e devidamente matriculado num curso, deverá acessar o “Espaço Meu” e clicar no curso no qual está matriculado.

**Exceções:**

Caso não exista matrícula, ocorre um erro.

5. **Evento:** Professor Autor deseja criar/manter páginas.



**Ator:** Professor Autor

**Descrição do fluxo:**

O professor autor poderá catalogar páginas desenvolvidas no banco de páginas, abstraindo-se da sua aplicação. Acionando o botão “Cadastrar Páginas”, é aberto um formulário para a criação/catalogação da página; deverá ser escolhida a categoria da página (conceito, fórum, teste, faq, atividade).

No caso de páginas do tipo fórum, teste, faq e atividade é necessário preencher informações adicionais de cada objeto. Em todos esses casos, o professor poderá fazer a manutenção de cada função desejada. As páginas criadas poderão ser utilizadas em diferentes cursos. É importante salientar que essas páginas, independente do seu tipo, podem ser incluídas mais tarde na estrutura de um curso (via autômatos). Assim, uma página de Fac, por exemplo, pode ficar vinculada a um determinado tipo de nó da hierarquia de conceitos, sendo tratada como um objeto a parte.

**Exceções:**

Caso algum campo importante no formulário da página não for preenchido, ocorre um erro.

No caso de exclusão de páginas do banco, poderá ocorrer erro se a página estiver vinculada a algum autômato.

A seguir é mostrada a interface para cadastramento de páginas.

## UNICLICK2000

prof  
Alterar Cadastro  
Espaço meu  
Mensagens  
Logout

**Curso ASP**  
Autor: prof  
Tutor: prof

Material Instrucional ▾

Mural | Chat | Galeria | Midiateca | Sumário | Tutoria | Situação do Aluno

**Sumário**  

- Curso de ASP (Incluir) (Excluir) (Concatenar)
  - Unidade 1 (Incluir) (Excluir) (Concatenar)
  - Unidade 2 (Incluir) (Excluir) (Concatenar)
  - Unidade 3 (Incluir) (Excluir) (Concatenar)

**+ Criação e manutenção de autômatos**  
**- Criação e manutenção de páginas**

Mostrar minhas páginas
Cadastrar nova Página

Mostrando todas as Páginas	Tipo	Ação
Curso ASP	texto	Remover   Editar
Unidade 1	texto	Remover   Editar
Objetivos de Aprendizagem	texto	Remover   Editar
Atividades Lição 2	atividades	Remover   Editar

**Cadastrando nova página**  
Título:   
Descrição:   
Conteúdo (aparece antes do arquivo composto):   
Para adicionar imagens: cadastre a imagem desejada na midiateca, pegue o código dela e utilize comando: imagem(cod, Descrição).  
Ex.: imagem(10, Teste)  
Arquivo Composto:  **Procurar**  
arquivo zip com um arquivo index.htm na raiz do arquivo  
Categoria: **Texto** ▾  
**Cadastrar**

Figura 22 – Interface para o professor autor na criação e manutenção de páginas, autômatos e estrutura do curso.

## 6. Evento: Professor Autor deseja montar estrutura de um curso

Ator: Professor Autor

### Descrição do fluxo:

Para montar a estrutura de um curso no ambiente Uniclick2000, o professor autor deverá seguir os seguintes passos:

- desenvolver o trabalho de pré-autoria, ou seja, deverá possuir toda a estrutura e material do curso preparada fora do sistema. Este material poderá ser desenvolvido em processador de textos e transformado em páginas no formato HTML (simples ou composto). Cuidados deverão ser tomados na segmentação do material instrucional, objetivando facilitar mais tarde a estruturação do curso (segmentos pequenos).
- realizado o trabalho de autoria, deverá catalogar as páginas no sistema no banco de páginas disponibilizando-as para a montagem dos cursos (evento cinco).
- deverá criar o autômato para suportar a estrutura do curso (evento oito);
- criado o autômato será feita a edição final do curso, através de ferramenta própria do sistema: a edição é a partir do nó raiz da árvore de conceitos, clicando a operação (inclusão, concatenação e exclusão de páginas) desejada no nó visualizado. Internamente são criadas as transições do autômato e vinculadas as páginas de saída (máquina de Moore/ máquina de Mealy).

Caso seja uma página de fórum, teste, atividade e faq, é necessário clicar na página desejada para adicionar os itens desejados. Em todos esses casos, o professor poderá fazer a manutenção de cada função desejada.

#### **Exceções:**

Caso a página já esteja no autômato atual, ocorre um erro.

7. **Evento:** Professor deseja corrigir atividade feita pelo aluno.

**Ator:** Professor tutor

#### **Descrição do fluxo:**

O professor, com o curso em andamento, acessa a página onde está a atividade desejada; neste local clica na atividade desejada e atribui a nota ao aluno.

8. **Evento:** Professor autor deseja manter autômatos.

**Ator:** Professor Autor

#### **Descrição do fluxo:**

Um autômato define a estrutura do modelo do domínio de um curso em termos de navegação, ou seja, através do autômato o professor pode criar a sequência do material instrucional desejada para o curso. Para um mesmo curso podem ser criados diferentes autômatos, cada um representando uma determinada sequência e conteúdo

do material instrucional. Por exemplo, um autômato para o curso normal, outro para revisão de assuntos de interesse da disciplina e assim por diante.

Ao clicar em **autômato** o usuário entra em formulário próprio para incluir, alterar ou excluir autômato.

Caso haja mais de um autômato para um determinado curso, o usuário poderá fazer a seleção na parte superior da tela onde consta “Material Instrucional”.

**Exceções:**

Caso algum campo esteja em branco, ocorre um erro. O processo de exclusão de autômato deve ser visto com atenção, pois poderá acarretar em exclusão também de itens relacionados. Por exemplo, avaliações, testes realizados no curso correspondente.

**9. Evento:** Professor tutor deseja manter mural.

**Ator:** Professor tutor.

**Descrição do fluxo:**

Para manter o mural, o professor deve clicar na opção mural, cadastrar mural e entrar com os dados necessários. Para remover o mural, é necessário ser o autor do mural e clicar em remover.

**Exceções:**

Caso algum campo esteja em branco, ocorre um erro.

**10. Evento:** O professor autor quer definir esquema de avaliação para um curso.

**Ator:** Professor autor.

**Descrição do fluxo:**

O professor autor poderá definir no início de um determinado curso, um esquema de avaliação, indicando para cada item (testes, atividades, participação em fórum) o peso que terá na avaliação geral do aluno no curso. As notas dos itens devem estar na faixa de 0 a 10. Definido o esquema de avaliação, o sistema poderá a qualquer momento mostrar o desempenho dos alunos no decorrer do curso. Para definir o esquema o professor aciona o botão próprio (esquema de avaliação) e aparecerão na interface todos os itens que podem ser avaliados – o professor preenche os pesos para o cálculo total do desempenho do aluno.

**11. Evento:** o professor autor quer criar testes de auto-avaliação.

**Ator:** professor autor.

**Descrição do fluxo:**

O sistema Uniclick2000 permite a elaboração de três tipos de testes com correção automática *on-line*: múltipla escolha, verdadeiro ou falso, coluna da esquerda com coluna da direita. Os testes podem ter quantos itens forem necessários. A inclusão de testes na estrutura do curso é feita como inclusão de página (evento cinco).

**12. Evento:** Aluno deseja publicar perguntas na tutoria

**Ator:** Aluno

**Descrição do fluxo:**

Para enviar perguntas ao professor o aluno utiliza a ferramenta tutoria para enviar as perguntas desejadas. Se a pergunta foi respondida, o ícone da tutoria fica diferente.

**Exceções:**

Caso algum campo esteja em branco, ocorre um erro.

**13. Evento:** Professor deseja responder dúvida do aluno pela tutoria

**Ator:** Professor

**Descrição do fluxo:**

Para responder a pergunta do aluno, o professor deve clicar na duvida desejada, preencher o formulário de resposta e clicar em enviar. Feito isso, a tutoria estará respondida.

**Exceções:**

Caso algum campo esteja em branco, ocorre um erro.

**14. Evento:** Usuário deseja efetuar logoff.

**Ator:** Usuário.

**Descrição do fluxo:**

Para sair do sistema, o usuário deverá escolher a opção efetuar Logoff que consta da interface principal.

**15. Evento:** Usuário deseja enviar/manter mensagens

**Ator:** Usuário

**Descrição do fluxo:**

O Ambiente possui um sistema interno de mensagens que possibilita enviar mensagens de um usuário para outro. Para enviar mensagens é necessário clicar na opção **Mensagens**, escolher o destinatário, preencher título e conteúdo e clicar em enviar. Feito isso, a mensagem será encaminhada ao usuário desejado. Caso o usuário receba alguma mensagem, a mesma irá aparecer na parte superior da tela.

**Exceções:**

Caso algum campo esteja em branco, ocorre um erro.

## **6 Modelo conceitual básico do modelo do domínio**

O modelo do domínio, mapeado para autômatos com saída (máquinas de Moore e Mealy), está representado nos diagramas ER abaixo. A entidade associativa denominada Composição torna explícita a relação entre página de saída e o alfabeto de saída, incluindo a condição de inclusão de fragmentos (páginas) para futuras implementações. A entidade Conceitos também está colocada para permitir uma relação explícita entre conceitos compostos (não terminais) e páginas de saída em futuras implementações de hipermídia adaptativa.

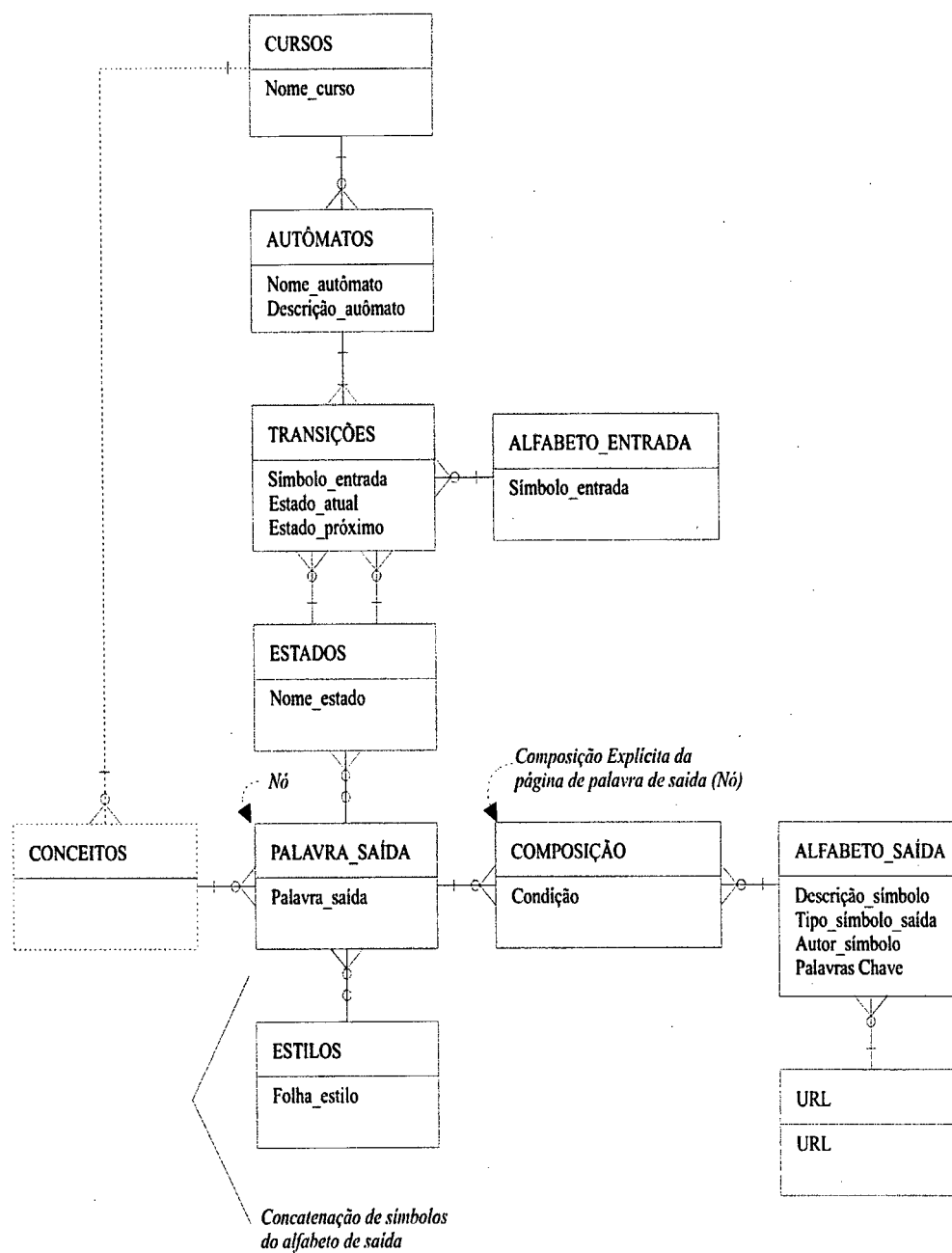


Figura 23 – Diagrama ER Máquina de Moore

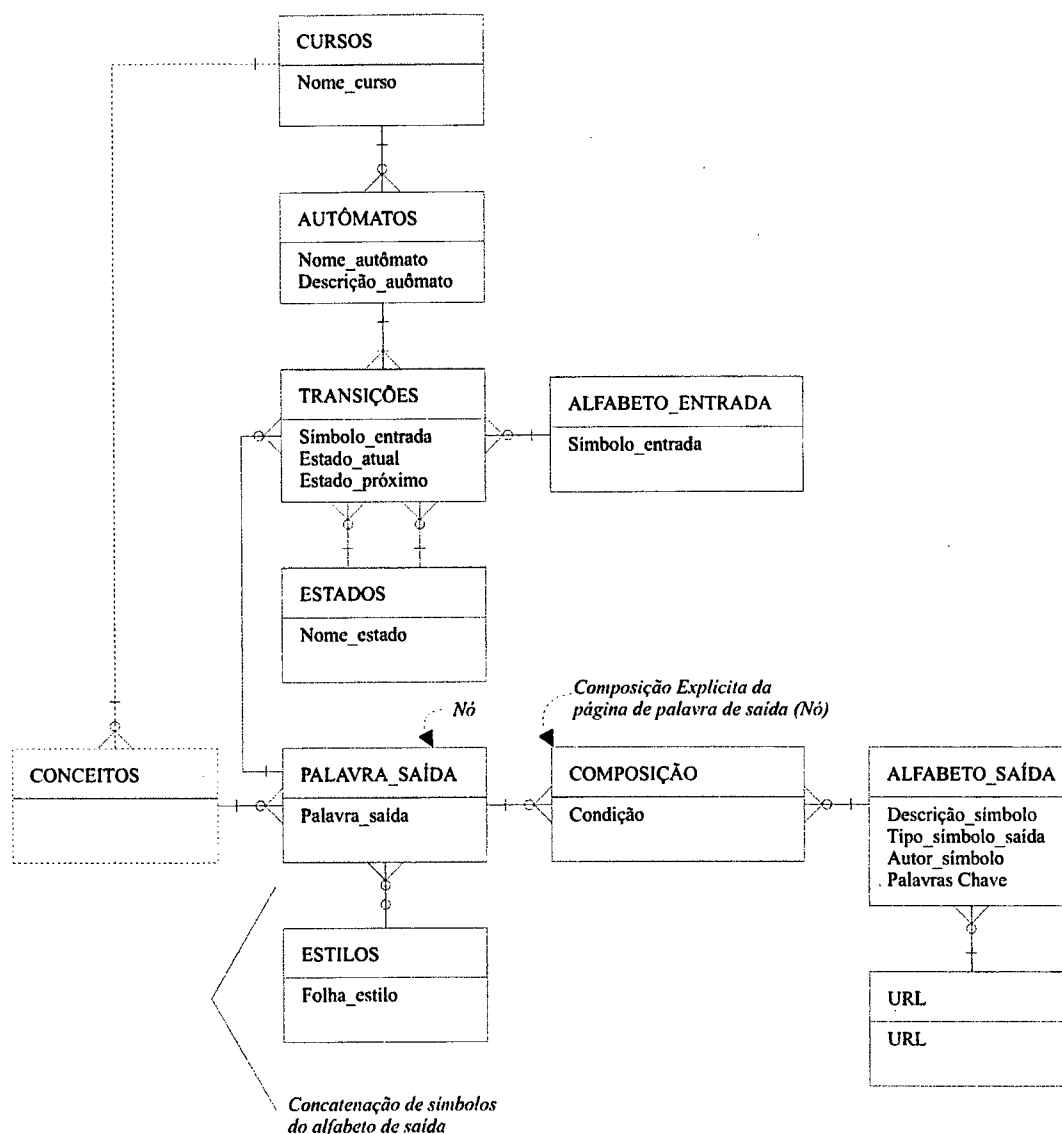


Figura 24 – Diagrama ER – Máquina de Mealy

## 7 Arquitetura computacional

O sistema foi projetado para funcionar na arquitetura em três camadas. No caso de uso do ambiente JAVA foram utilizados recursos das API's Servlets e RMI, já referenciadas no capítulo sete.

A figura 25, a seguir, mostra os elementos da arquitetura. Foram utilizados duas máquinas: a primeira, comportando o servidor Web e o servidor de aplicações, em ambiente Linux; a segunda máquina com um servidor de banco de dados relacional em ambiente NT.

No servidor de aplicações estão as regras do negócio, no caso todos os módulos



(objetos) projetados para o sistema, como autômatos, banco de páginas, ferramentas de comunicação etc.

No servidor de banco de dados estão todas as tabelas projetadas para o sistema.

O módulo de interface foi construído utilizando recursos da API Servlet (classe `http-servlet`) e funciona junto ao servidor Web. Nessa arquitetura, quando uma requisição do usuário sobre o material instrucional de um determinado curso chega ao servidor Web, ela é repassada para o módulo de interface para processamento. Este módulo faz a validação imediata e envia os dados para o módulo Autômato que está no servidor de aplicações. Com base na matriz de transição do curso em pauta e função de saída correspondente, o módulo Autômato requisita ao servidor Web os documentos (páginas) necessários e constroi a página para ser apresentada ao usuário.

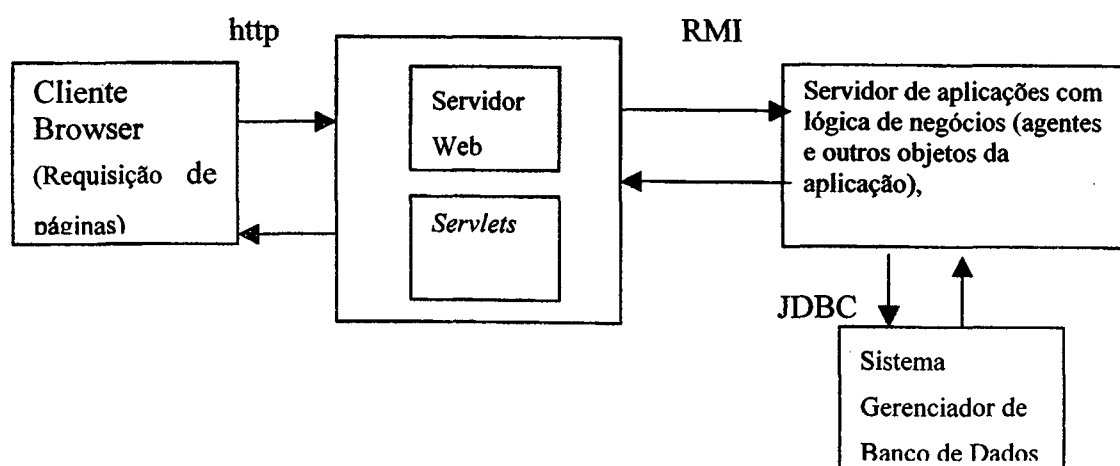


Figura 25 – Arquitetura computacional utilizada no protótipo.